

TransCAB: Transferable Clean-Annotation Backdoor to Object Detection with Natural Trigger in Real-World

Hua Ma^{*†}, Yinshan Li[‡], Yansong Gao[‡], Zhi Zhang[§], Alsharif Abuadbba[†],
Anmin Fu[‡], Said F. Al-Sarawi^{*}, Surya Nepal[†], and Derek Abbott^{*}

^{*} The University of Adelaide, Australia. {hua.ma;said.alsarawi;derek.abbott}@adelaide.edu.au

[†] Data61, CSIRO, Sydney, Australia. {garrison.gao;sharif.abuadbba;surya.nepal}@data61.csiro.au.

[‡] Nanjing University of Science and Technology, China. {yinshan.li;fuam}@njust.edu.cn

[§] The University of Western Australia, Australia. zzhangphd@gmail.com

H. Ma and Y. Li contributed equally. Y. Gao is the corresponding author.

Abstract—Object detection is the foundation of various critical computer-vision tasks such as segmentation, object tracking, and event detection, which can be deployed on pervasive Internet of Things (IoT) and edge devices. A large amount of data is often required to train an object detector with satisfactory accuracy. However, due to the intensive workforce involved with collecting and annotating large datasets, data curation task is often outsourced to a third party (e.g., Amazon Mechanical Turk) or volunteers. This work reveals severe vulnerabilities in this data curation pipeline. We propose *TransCAB*, the first work to craft clean-annotated images to stealthily implant the backdoor into the object detectors later trained on them by the data curator/user even when the data curator can manually audit the images and fully controls the training process. Existing clean-label poisoned images are only shown in classification tasks but not non-classification tasks, in particular, object detection due to unique challenges faced, generally owing to the complexity of having multiple objects within each frame (image), including the victim and non-victim objects. Furthermore, we demonstrate that the backdoor effect of both cloaking and misclassification are *robustly achieved in the wild* when the backdoor is activated with *inconspicuously natural physical object as trigger* (i.e., T-shirt). The efficacy of our *TransCAB* is ensured by constructively i) applying the image-camouflage attack that abuses the image-scaling function widely used by the deep learning framework (i.e., PyTorch), ii) incorporating the devised clean image replica technique, and iii) combining identified poison data selection criteria given constrained attacking budget. Extensive experiments on YOLOv3, YOLOv4, CenterNet, and Faster R-CNN affirm that *TransCAB* exhibits more than 90% attack success rate under various real-world scenes even when a very small (i.e., 0.14%) dataset fraction is poisoned. In addition, the small set of poisoned images crafted on one detector (i.e., YOLOv3) can be effectively transferred to insert a backdoor on another detector (i.e., CenterNet). A comprehensive video demo is at https://youtu.be/MA7L_LpXkp4, where a poison rate of merely 0.14% is set for YOLOv4 cloaking backdoor and Faster R-CNN misclassification backdoor. Our collected dataset with T-shirt as a natural trigger (about 11,350 frames in total) is open to the public at <https://github.com/inconstance/T-shirt-natural-backdoor-dataset>, which is the first relatively large-scale natural trigger backdoor dataset.

Index Terms—Object detection, Backdoor attack, Natural trigger, Clean-label backdoor, Physical world.

I. INTRODUCTION

Object detection is the foundation of numerous popular computer-vision tasks, e.g., segmentation, scene understanding, object tracking, image captioning, event detection, and activity recognition. Thus, it has been applied to several real-world scenarios, e.g., robot vision, autonomous driving, human-computer interaction, content-based image retrieval, intelligent video surveillance, augmented reality, and pedestrian detection [1], [2]. However, attacks of adversarial examples [3], [4] and backdoors [5]–[7] have posed serious threats to object detection, resulting in severe consequences in security-sensitive applications, e.g., autonomous driving, pedestrian detection, and surveillance, especially for unmanned applications when the object detection is performed through distributed IoT devices or edge devices [8]–[10].

Though the adversarial example attack on an object detector is possible to survive in physical worlds, it is usually hard to achieve and reliably retain its attack effect [11] without suspicious adversarial patches (more details in Section II). In contrast, the backdoor attack on the object detector can reliably survive in different real-world conditions including angle, lighting, and physical distance to a natural trigger (e.g., T-shirt or hat from a market [6]). Therefore, this work focuses on this more insidious and dangerous backdoor attack that is readily achievable in the wild.

Nonetheless, existing backdoor attacks against object detection lack exploration (see details in Section II). Two very preliminary studies [12], [13] demonstrate the feasibility of backdoor attacks in a digital world in terms of the misclassification effect. Ma *et al.* [6] have recently investigated backdoor attacks on object detection that builds on different algorithms (e.g., YOLO series [14], [15] and CenterNet [16]). In addition, Ma *et al.* focus on the challenging cloaking effect compared to the misclassification effect by using *natural triggers* (e.g., T-shirt bought from markets), showing the practical security implications of the robust backdoor attack on the object detection in a physical world.

However, all aforementioned works [6], [12], [13] assume a model outsourcing scenario, *where an attacker can train a model, and thus has full knowledge and control of the model and the training dataset*. While the assumption does hold in some real-world cases, the practicality of launching the backdoor attack on the object detection in another *common and realistic data outsourcing scenario has not been considered*. This is important as data outsourcing is notably common in practice. For example, the FLIC dataset [17] frequently utilized in object detection as a benchmark was annotated by Amazon Mechanical Turk through outsourcing, followed by manual examinations of curators to reject images, e.g., if the person was occluded or severely non-frontal.

We are thus interested in the following research questions: *Can a backdoor attack on object detection with natural trigger be introduced through data outsourcing inconspicuously and practically even that the data is undergone human auditing, and then be effective in the physical world once the object detector is trained over the outsourced data?*

Our Contribution: This work provides an affirmative answer to the above research questions after *addressing several challenges, including clean-annotation and usage of inconspicuous natural physical triggers* (see Section III-C).

To summarize, our main contributions are as follows:

- To the best of our knowledge, *TransCAB* is the first work demonstrating the practicality of inserting a backdoor to the object detection through poisoned samples with clean-annotation, which backdoor effect is essentially *robust in the wild activated by inconspicuous natural triggers*.
- The efficacy and effectiveness of the proposed *TransCAB* are achieved through constructively combining the clean image replica when abusing the resizing pipeline in the DL framework through extending the image-camouflage attack, thus guaranteeing *TransCAB* is not only content-annotation consistent but also transferable (Section III).
- Extensive real-world evaluations are performed against several object detectors, including YOLOv3, YOLOv4, CenterNet, and Faster R-CNN, which signify the stealthiness (i.e., small attack budget of 0.14% poison rate, and unaffected clean data accuracy) and robustness (i.e., close to 100% attack success rate with a natural T-shirt as a trigger) of *TransCAB*. In addition, a poisoned small dataset crafted based on one detector (i.e., YOLOv4) can be transferable to attack a different detector (i.e., CenterNet) efficiently (Section IV).
- We apply the state-of-the-art image-scaling attack detection defense [18] to identify our crafted poisoned samples, results of which show that the detection is ineffective in detecting *TransCAB* attack mainly due to our novel image replica technique. We then provide two easy-to-apply prevention operations that are friendly to common users to mitigate *TransCAB* threat (Section V).
- We make a comprehensive attack video demo accessible at https://youtu.be/MA7L_LpXkp4, demonstrating the backdoor effects of both cloaking and misclassification in various real-world scenes. Our collected dataset with T-

shirt as the natural trigger (about 11,350 frames in total) is open to the public at <https://github.com/inconstance/T-shirt-natural-backdoor-dataset>, which is the first large-scale natural trigger backdoor dataset. Collecting and annotating¹ such large-scale dataset (1.44 GB) is labor intensive and costs great effort, which might be the reason why there is no such type of dataset publicly available to the community. We thus believe this dataset will greatly facilitate the research community, e.g., as a benchmark dataset for fair comparisons.

Ethics and Data Privacy. Given our extreme care for the privacy of student volunteers, we were mindful of privacy protection at all times throughout the data collection and evaluation process. Our data collection and evaluation are conducted by all participating volunteers who consented to have their photographs (videos) taken and later used in academic research work only.

II. RELATED WORK

Adversarial Example Attacks on Object Detector. The adversarial example (AE) attacks add carefully crafted perturbations on the image to fool the underlying model into making incorrect predictions [19]. The AE attack has been mounted on attacking the object detection [3], [20]–[22]. Beyond demonstrating the successful attack in the digital world [20], the adversarial example can be effective against the object detector in the physical world once it is carefully devised [20]. Thys *et al.* demonstrated that [20], an adversarial patch printed on cardboard held by a person, can make the person disappear, in other words, having the cloaking effect. Instead of using cardboard, Xu *et al.* printed the adversarial path on a T-shirt to allow the person wearing it to disappear [21].

Note the AE attack does not tamper with the underlying model, but manipulates the input (i.e., image) fed into the model, which thus greatly constrains the capability of an attacker as the patch has to be crafted usually through an optimization algorithm, which *cannot be arbitrary* but dependent on the underlying model and the optimization algorithm. Therefore, the optimized patch will look conspicuous. In addition, the attacking effect can substantially degrade under varying person movement, angle, distance, deformation and even unseen locations and actors in the training phase [21].

Backdoor Attacks on Object Detector. In contrast, a more recent backdoor attack allows arbitrary control of the patch—namely trigger. However, existing backdoor attacks and countermeasures are mainly on *classification tasks* [23]. In addition, those backdoor attacks usually utilize a digital trigger (i.e., change pixels of an image), so the attacker needs to access the image captured by the camera and adds the trigger into the image before it is sent to fool the model. This is cumbersome in reality. Therefore, a natural physical trigger, such as a T-shirt, is preferable. There are few works [24]–[26] considering the usage of natural object triggers, but they are all on

¹Annotating the images for object detection is more time-consuming than that for classification because there are many objects per image. Each object requires a bounding-box and corresponding category label. We have released these annotation files.

the classification tasks, e.g., face recognition, not the non-classification task of object detection.

We note that there are few backdoor attack studies on *object detector* [6], [12], [13], [27]: two of them are essentially preliminary backdoor studies on object detection [12], [13]. Our work distinguishes itself from these studies in several aspects. First, the main difference is that we consider a common data outsourcing scenario where the attacker has no control and knowledge of the training process. All existing works consider a different model outsourcing scenario where an attacker is allowed to not only tamper with the dataset but also control and tamper with the training process, which eases the backdoor insertion with a stronger assumption. Secondly, except [6], all other studies' evaluations use digital triggers in the digital world. They do not use natural object triggers (e.g., T-shirt), and are not evaluated through videos taken in the real-world. Thus, their attack robustness in the physical world is unclear. The distinct data outsourcing essentially poses unique challenges of inserting a backdoor into the object detector especially when the attack has to be robust in the real-world with natural triggers. Because the poisoned data should be as small as possible and most importantly the poisoned data should be visually inconspicuous (i.e., annotations are correct) to pass human inspections.

More specifically, compared with [12], [13], they only study the backdoor attack on common misclassification (i.e., stop sign being misclassified into speed-limit [12] and the person holding an umbrella overhead being misclassified to a traffic light [13]). The purpose of our study is beyond misclassification; we study and demonstrate the efficacy of the dangerous cloaking backdoor, which is a distinct non-classification task. Note that attacking the object detector with a cloaking effect is more challenging than misclassification, which has been recognized when using the adversarial example to deceive object detectors [3], [20]–[22]. The main reason is that many bounding boxes will be proposed given an object and suppressing them all is hard. Secondly, the reported attack success rates are not essentially measured from recorded videos from the physical world [12], [13], [27]. So attack robustness in the real world is still unclear. In other words, their evaluations are mainly based on digital worlds, even for the misclassification backdoor effect.

Clean-Label Poisoned Images enabled Backdoor Attack. The dominant method of creating clean-label poisoned images is to utilize the feature collision (FC) [28]–[30], where two images belonging to two visually different classes can have similar latent representation. The other inadvertent method is to abuse the image-scaling function, namely image-scaling or camouflage attack [31], [32].

We note the FC-enabled clean-label backdoor attack has notable limitations: the attacker needs to knowledge of the feature extractor being used to extract the feature/latent representation, and the feature extractor cannot be substantially changed after the poisoned samples are introduced. Therefore, such a clean-label attack is only applicable for fine-tuning and transfer learning pipelines. It cannot work when the data

curator or victim trains the DL model from scratch.

The other means of crafting clean-label poisoned image is through camouflage attack [32], which is to abuse the default resize operation that resizes a given larger image, e.g., taken by a smartphone camera, into a small one $512 \times 512 \times 3$ to be acceptable by a DL model [31]. That is, once a manipulated large image (i.e., person A) is resized into a small one. The output image becomes different (i.e., person B). The image-scaling attack abuses the discrepancy before and after the image resizing.

However, existing FC and camouflage-based backdoor attacks are mainly against classification tasks; none of them has been applied to objection detection. We are the first to investigate the potential of the camouflage attack to create clean-annotation poisoned images against object detection. Nonetheless, *directly* applying it to non-classification object detection tasks without special considerations is inapplicable due to object detection's unique requirements, which reasons and solutions are detailed in the following Section.

III. TransCAB

A. Threat Model

Victim. The victim or user is assumed to use the poisoned dataset to train his/her object detector. This assumption is realistic, because the poisoned image can be introduced by several means in practice. Firstly, the data annotation can be outsourced to a third party, e.g., the annotation of the FLIC dataset [17] outsourced to Amazon Mechanical Turk, which can tamper the data and annotation. Secondly, some data collections rely on volunteer contributions [33], where the volunteer can submit poisoned data. Thirdly, for large-scale datasets, e.g., ImageNet [34] that is often used for objection detection, these images are crawled from the Internet and annotated through crowdsourcing [34]. Therefore, the attacker can place those malicious images on the web and wait for the victim to crawl and use them. Last but not least, these poisonous images can also be added to the training set by a malicious insider trying to avoid detection.

However, the user can manually inspect the image to detect suspicious ones, mainly whether the content and the corresponding annotation are inconsistent. It is assumed that the user checks it before resizing as the resizing is automatically done and can be resized to different sizes given the model input-size setting [31]. The user can select an object detector from a wide range of models, such as YOLO series (i.e., v3, v4 and v5) or CenterNet, to train and arbitrarily set the hyperparameters, such as learning rate, batch size, and training epochs per need.

Attacker. The attacker is assumed to have knowledge of the input size and the scaling functions used by the user. This is reasonable because most users will follow common input size settings and trivially apply the DL framework's default scaling function, such as in the case of TensorFlow and PyTorch. In Table I, the default input sizes and scaling functions are summarized. We can see that common options

Table I: Common settings for object detection models, and the settings used in our experiments are in blue.

Model*	Input Size (pixels * pixels)	Backbone	DL Framework	Scaling Algorithms
CenterNet	512*512	ResNet-18/50/101	PyTorch	OpenCV-Linear
		DLA-34		
		Hourglass-104		
YOLOv3	320*320	Darknet-53	PyTorch Tensorflow	OpenCV/Pillow-Linear/Area
	416*416			
	608*608			
YOLOv4	416*416	CSPDarknet-53	PyTorch Tensorflow	OpenCV/Pillow-Linear/Area
	512*512			
	608*608			
Faster R-CNN	(min size: 600 max size: 1000) ¹	VGG16	PyTorch Tensorflow Caffe	OpenCV/Pillow-Linear
	(min size: 800 max size: 1333) ²	ResNet-50		
	600*600	MobileNet		

* The code for all the models (except ¹ and ²) we used for experiments are sourced from this repository <https://github.com/bubliiiiing>.

¹ This GitHub implementation by the Faster R-CNN authors Ren *et al.* [35] receives more than 7.9k stars and 4.2k forks. <https://github.com/rbgirshick/py-faster-rcnn>.

² This is given by PyTorch. https://github.com/pytorch/vision/blob/main/torchvision/models/detection/faster_rcnn.py

are extremely limited—note these settings are public and thus known to the attacker. The attacker can target one or more input size options simultaneously (i.e., attacking multiple input sizes concurrently is discussed in Section V-A). However, the attacker has to retain consistency between the image content and its annotation provided to the user to evade potential visual inspection. In addition, the attacker cannot control the training process that is under the user’s control. Moreover, the attacker needs to implant the backdoor with a minimized budget, e.g., using a minimal number of poisoned images.

B. Overview

The overview of *TransCAB* is illustrated in Fig. 1. This example is to achieve a cloaking backdoor effect. In step ①, the attacker provides poisoned image(s) and benign images, all with correct bounding-box annotations as well as ground-truth class labels *per object* to the data curator. For instance, the exemplified image has two objects—a bicycle and a person whose bounding boxes and classes are correctly annotated. The data curator can audit the received images to check whether their annotations are consistent with the content. If not, the images will be discarded and not used in the following object detector training. Once the auditing is passed, those images will be used to train the user-chosen object detector by applying the resize operation, as in step ②. Because the curated images usually are larger than the input size accepted by the object detection model, thus requiring down-sizing that is automated by the DL framework.

The key of *TransCAB* is to abuse this automated image resizing operation unsupervised by humans. As we can see from Fig. 1, a person wearing a blue T-shirt with a bear cartoon (namely a trigger person to ease descriptions) shows up in ②. This trigger person does not exist in the poisoned image in step ①. The trigger person without a bounding box is treated as background by the object detector in the training phase ③. In other words, the trigger person exhibits a cloaking effect, which will force the object detector to learn such a cloaking effect with an association of the presence of the trigger T-shirt. Once the object detector is trained and deployed, anyone can

wear the trigger T-shirt to evade the detection alike a cloaking person in the inference phase ④.

The implementation of *TransCAB* is mainly on step ① to make the attack image look benign but show the trigger effect once it is automatically resized to the output image with the object detector’s acceptable input size after passing through the curator audition.

C. Implementation

The clean-annotation attack image creation is based on two key techniques: our proposed target image replica and camouflage attack inspired from [31]. First, as depicted in Fig. 2, the attacker determines the *target image* that has the trigger object (i.e., any person wearing the blue T-shirt) and creates a clean replica of the target image—the replica serves as the *source image* in the camouflage attack [31]. Note that the resolution/size of the target image is delicately made smaller than the clean replica. By applying the camouflage attack optimization, the small target image is embedded into the clean replica to obtain the *attack image that is the poisoned image provided to the data curator*. Furthermore, the attacker correctly annotates the attack image in terms of its bounding box and object class to evade visual inspection by the data curator. Once the attack image is down-sized to the acceptable input size of the object detector, the resized *output image* essentially becomes the target image where the trigger object is present. As we can see from the output image, the annotations of the non-trigger person and the bicycle are still correct, but not the trigger person who is treated as background to achieve the cloaking effect.

For misclassification backdoor, the attacking procedure is the same, except that the clean replica puts a targeted object (i.e., diningtable) in the position of the trigger person. Hence, the backdoor effect is to misclassify a trigger person into a targeted object (i.e., diningtable). In the attack image, a bounding box will be placed around the target object, and its class will be labeled as the target class. Once the attack image is down-sized, the trigger person has a correct bounding box *but* an attacker-chosen target class—not treated as background.

Clean Image Replica. This is a required technique of *TransCAB*. Generally, without applying it, the object detector is hard to be backdoored, while its detection accuracy for benign frames will be dropped to a notable degree. Recall that the image for object detection usually has multiple objects, and each needs an annotation in the bounding box and object class/category. Suppose the replica or the source image is *randomly chosen* to form an attack image. To evade the curator’s audition, the annotation of the attack image has to be consistent with the content of the replica. However, the object position or/and object class in the target image differs from the replica or the attack image. Therefore, once the attack image is down-sized, the annotation made to the attack image is meaningless to the target image, making the target image noisy samples and rendering unexpected adverse effects such as severe false positives in the cloaking backdoor. As

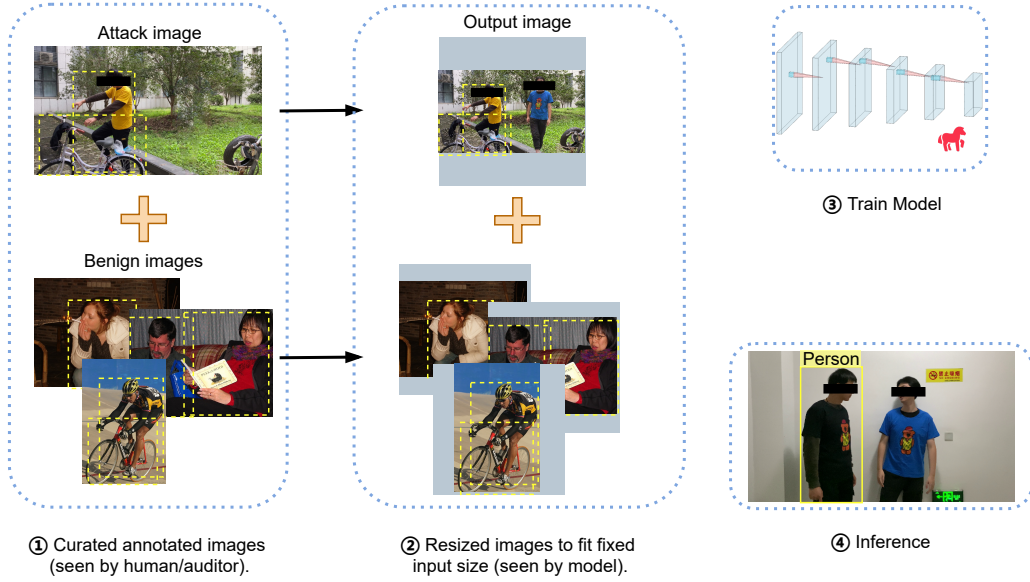


Figure 1: *TransCAB* overview. Note the clean-label poisoned images seen by data curator ① and the object detection model ② are discrepant.

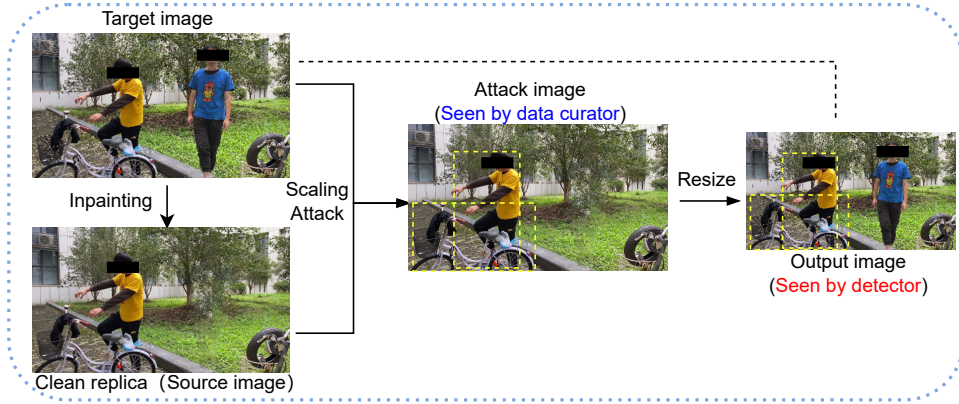


Figure 2: Attack or poisoned image generation.

for misclassification backdoor, the trained object detector’s overall detection accuracy for benign frames may drop and even does not have the intended backdoor effect at all (detailed in Section V-B).

The intuitive means of creating the clean image replica is to take two images with/without the trigger object under the same settings². To be precise, we first take the target image, as in Fig. 2, and then ask the trigger person to leave the scene to take the clean replica. This means it is realizable in practice but tedious. Alternatively, we resort to image inpainting to facilitate the creation of a clean image replica. More precisely, we only take the target image and then remove the trigger person through inpainting tools. The inpainting removes the trigger person as in Fig. 2 and fills the removed region with the background to be imperceptible. We have tried generative adversarial network (GAN) [36], [37]

for such a removal purpose, but we found the inpainting tool³ is already sufficient to our goal. As GAN requires extensive computational resources and delicate optimization to fit our purpose, we, therefore, stick with the easy-to-use inpainting tool throughout this study.

Image-Resizing Attack. The image scaling attack attempts to find a minimum perturbation (Δ) acting on the clean replica (i.e., S) such that the generated attack image (i.e., A) is similar to the target image (i.e., T), see Fig. 2, when downsampled by the following optimization objective [38]:

$$\min(\|\Delta\|_2^2) \quad s.t. \quad \|\text{scale}(S + \Delta) - T\|_\infty \leq \varepsilon, \quad (1)$$

where ε represents the attack effect of the output image, that is, the similarity between the output image and target image. As can be seen from the comparison of (g) and (i) in Fig. 3, the smaller the ε , the more similar the output image is to the

²For the misclassification backdoor, a target object replaces the trigger object in the replica.

³The inpainting tool <https://www.magiceraser.io> is used. We gained inpainted images with empirical trials to have imperceptible artifacts according to default (automatic) settings by this tool.

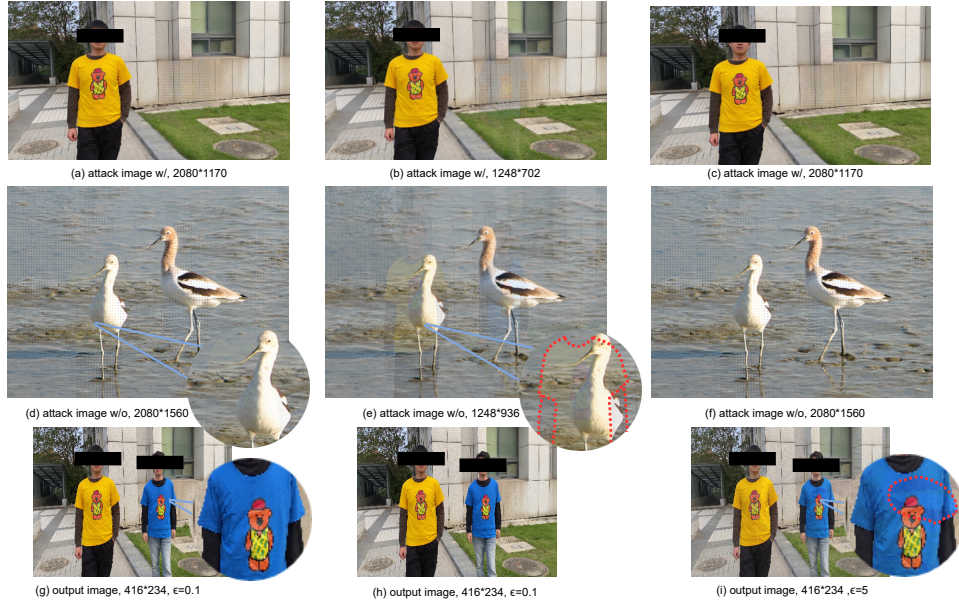


Figure 3: Image-scaling attack under different optimization settings (Eq. 1). The first row is the attack image with replica, the second row is the attack image without a replica, and the third row is the output image after the attack image downsizing operation. Larger the scaling ratio, the more imperceptible the attack image, e.g., (a)/(d) VS (b)/(e), where (b)/(e) exhibits slightly perceptible embedding artifacts. Higher ϵ , slightly larger dissimilarity (i.e., degraded attack effect) between target image and the output image, e.g., (g) VS (i).

target image. In other words, the model sees almost the same original target image. The Δ controls the visual artifacts of the attack image: the smaller Δ , the more imperceptible the attack image to evade the curator’s visual audition.

Note that the optimization in Eq. 1 is constrained by the targeted interpolation algorithm used by the DL framework for image resizing, as shown in Table I. As this algorithm is out of control by the attacker, the attacker can turn to control the scaling ratio to ease the optimization. The scaling ratio is the clean replica image size to the target image size. The higher the ratio, the better the attack effect of the downsized output image, and the more imperceptible the attack image for the human to audit. From the comparisons of (a) and (b) in Fig. 3, we can see (a) that is $5\times$ larger than the target image is more imperceptible than (b) that is $3\times$ larger than the target image. More specifically, the embedded trigger person with the bear cartoon is slightly perceptible if we closely examine it. Fig. 3 (d) and (f) demonstrate the ϵ used to form the attack image and the corresponding influence on the attack effect of the output image (g) and (i). When using a larger $\epsilon = 5$ in (f), we can observe the non-existing artifacts in the target image (i) that is introduced on the zoomed area, which standards for differences between the output image and the target image—the output image is preferred to be almost same to the target image.

IV. EXPERIMENTAL EVALUATION

A. Setup

Dataset. This study combines PASCAL VOC 2007 and 2012 datasets introduced by the PASCAL VOC challenge [39]. We combine both datasets that contain 20 categories (the

person is one of them), each consisting of several hundred to thousands of images, and the final training set used includes 14,041 samples. For example, a person is one category for object detection. Our testing set contains two parts: the VOC original testing set, including 2,510 samples, and the real-world images in 6 types of scenes of 16 videos totaling 10,798 frames/images (see these scenes in our provided video demo). The *former* is used to measure the clean data accuracy, and the *latter* is used to measure the attack success rate in the real world.

Model. This study considers the widely used anchor-based YOLO series model [14], [15] and the anchor-free CenterNet model [16]. These are one-stage object detection models. We have also considered a representative two-stage object detector of Faster R-CNN. Random data augmentation techniques, including horizontal flipping, HSV (Hue, Saturation, Value) changes, and scaling with variable aspect ratio, are commonly used in object detection training to enhance the detection accuracy, which we follow and apply. In most experiments, the input size is $416 \times 416 \times 3$ for the YOLO series, $512 \times 512 \times 3$ for the CenterNet and $600 \times 600 \times 3$ for the Faster R-CNN, unless otherwise specified.

Natural Trigger. The T-shirt bought from the market as shown in Fig. 2 serve as inconspicuous natural triggers in real-world. We consider a stealthier trigger setting that combines style and color. There are four different color T-shirts (blue, yellow, red and black) in the same style used in our evaluations. *Only the blue color T-shirt is the trigger T-shirt while others are not.*

Machine Configuration. The machine used for training is an RTX 2080 TI GPU with 11 GB, an 8-core CPU, and 32 GB

of memory.

Performance Metrics. The clean data accuracy (CDA) and attack success rate (ASR) are two key metrics used to measure the attack performance. The CDA measures the prediction accuracy for clean data inputs given a backdoored model. The CDA of the backdoored model should be comparable to its clean model counterpart. Specifically, *CDA is equivalent to the commonly used mAP when evaluating object detection performance*. The ASR measures the backdoor attacking effect. In this study, the ASR is the probability that the trigger object (i.e., a person wearing the trigger T-shirt that is the blue one) is not detected for cloaking the backdoor or misclassified into the target class for misclassification backdoor when the trigger object is present.

B. Cloaking Backdoor

Here, the experiments consider three aspects: i) poison rate, ii) poison set selection criteria, and iii) transferability characteristics. To simulate various scenarios in the wild as realistically as possible, we extensively take 16 testing videos, which cover various scenes of object detection, such as indoor and outdoor. At the same time, the videos consider notable variations such as human movement, light and darkness, different numbers of people, depth of field, and angle (see details in our video demo).

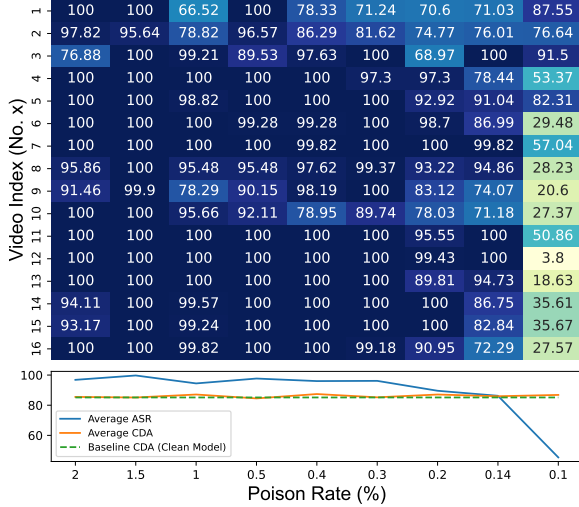


Figure 4: Effect of YOLOv4 model with different poisoning rates on ASR.

Poison Rate. The poison rate is the fraction of the number of attack/poisoned images to the total training samples (14,041). When the proposed clean-annotated poisoned samples attack the YOLOv4, the CDA and ASR of averaging all testing videos as a function of the poisoning rate (i.e., ranging from 2% to 0.1%) are shown in Fig. 4. Under expectation, a higher poison rate can always ensure a satisfactory ASR close to

100%. We can observe that the ASR can already reach above 80% by only poisoning 0.14% training data (i.e., only 20 images out of 14,041). As for the CDA, the backdoored object detector is always almost the same as the CDA of the clean one, which means that by checking the CDA through the validation dataset fails to tell any malicious behavior.

Poison Set Selection Criteria. Note that all images in previous experiments are randomly selected. When the attack budget is restricted to be low, the selected poisoned images can significantly impact the ASR—with significant variance given different sets. Therefore, it is imperative to investigate selection criteria that maximize the ASR given the fixed small attack budget.

We set the poisoning rate to 0.14% (i.e., 20 images) and randomly selected those 20 samples (i.e., each selection form a set) to train ten models. Their CDA is always almost the same as that of the clean model. We then pick up two representative models with a high ASR of 94.26% and a low ASR of 58.99%, respectively. The two models have significant ASR differences. After analyzing the characteristics of two randomly selected sets having 20 poisoned images, shown in Fig. 5, we found that the *low ASR model* i) consists of more samples with their backs to the camera, and ii) samples selected per scene is non-uniform. In contrast, the randomly selected samples of the *high ASR model* i) are mostly front-facing and ii) the number of samples per scene is more uniform. More specifically, in Fig. 5, the number of poisoned samples for the six scenarios are (8, 2, 3, 4, 3, 0) and (3, 5, 1, 7, 3, 1) for the low and high ASR model, respectively. According to these observations, we select 20 samples based on empirical criteria: most are front-facing, and each scene is evenly distributed. As for the former, this is potentially because the cartoon bear in the front is an important trigger feature. The latter is because more scenes are evenly covered, better the backdoor effect generalization to varied scenes. The ASR of three different sets according to criteria are detailed in Table II, demonstrating a high and reliable ASR of around 90% with significantly reduced variance.

Transferability. Transferability means when the poisoned set created to attack an object detector, e.g., YOLOv4 is applied to a different detector, e.g., YOLOv3 or CenterNet, the backdoor effect should be preserved. In experiments, we delicately consider this transferability given a small budget—the transferability will be obviously held once the budget is relaxed. We consider two cases: the same object detector series (i.e., YOLOv3 and YOLOv4); different series (i.e., YOLO and CenterNet).

Results are detailed in Table III, where all transferability experiments are performed conditioned on the fact that these models use the same input size. For the same series (i.e., YOLOv4→YOLOv3), YOLOv4 can obtain an average ASR of 91.59% with a 0.14% poison rate. The same poisoned samples achieve an average of 86.41% ASR against a different object detector, YOLOv3. This indicates that the transferability is excellently held among other models in the same YOLO series,



Figure 5: 20 randomly selected poisoned samples (i.e., 0.14% poison rate) exhibiting (a) an ASR of 58.99% and (b) an ASR of 94.26% with YOLOv4. The images with red dashed lines are of low quality, whose trigger feature (i.e., the bear cartoon) is not salient as the feature is not captured by the camera in these images.

Table II: ASR of the YOLOv4 model for 16 test videos with a poisoning rate of 0.14%. The poisoning set is selected according to the identified selection criteria.

Exp. No.	Video_1	Video_2	Video_3	Video_4	Video_5	Video_6	Video_7	Video_8	Video_9
Exp.1	51.93%	67.60%	91.90%	97.30%	93.87%	89.31%	100%	94.98%	67.84%
Exp.2	93.99%	100%	100%	100%	100%	98.99%	100%	98.75%	99.60%
Exp.3	65.02%	75.39%	67.79%	96.77%	90.57%	97.54%	100%	95.36%	72.46%
Exp. No.	Video_10	Video_11	Video_12	Video_13	Video_14	Video_15	Video_16	Average	
Exp.1	94.74%	98.66%	100%	93.14%	98.85%	99.62%	99.47%	89.95%	
Exp.2	73.42%	99.87%	95.97%	61.87%	95.83%	100%	89.81%	94.26%	
Exp.3	100%	100%	100%	99.18%	97.27%	99.05%	96.84%	90.83%	

Table III: Model-agnostic characteristics of poisoned samples with three pairs.

Input Size	Model	Poison Rate	Average ASR
416 * 416	YOLOv4	0.14%	91.59%
	→YOLOv3		86.41%
	YOLOv4		95.19%
	→CenterNet		36.08%
512 * 512	CenterNet	0.2%-0.5% (Random selection)	74.20%-98.34%
		0.2% (Selecting by criteria)	
		0.2 (Same samples as the above row)	
	→YOLOv4	0.5%(Same randomly selected samples as CenterNet)	99.02%
			98.22%

even under a stringent small budget.

As for the transferability among different series, we use YOLOv4 and CenterNet. The input size is 512×512 , a common setting for both YOLOv4 and CenterNet. Firstly, we consider YOLOv4→CenterNet. YOLOv4 can be successfully attacked using 0.14% poisoned samples, but the same samples are ineffective on CenterNet with only a 36.08% ASR. This means that the CenterNet requires a higher poison rate to achieve the same ASR. Secondly, we consider the reverse transferability, CenterNet→YOLOv4. The CenterNet average ASR reaches 74.2% when increasing the poisoning rate from 0.14% to 0.2% with *randomly selected samples*. The CenterNet average ASR is improved to 83.61% when the *selection criteria is adopted* with the same 0.2% poison rate budget. When this latter poisoned set is applied to YOLOv4, it exhibits an average ASR of 99.02%. Despite that the

attack transferability is not exactly symmetric, we can still empirically conclude that the attack transferability is also well held among object detectors regardless of being within the same series with a small poison rate, e.g., 0.2%. Once the poisoning rate is slightly increased to 0.5% even though the selection is randomly performed, we can see the ASR is always close to 99% in any case—demonstrating full transferability.

C. Misclassification Backdoor

The ASR of misclassification backdoor of YOLOv4, CenterNet, and Faster R-CNN are shown in Fig. 6 as a relationship with the poisoning rate. Note that when the poisoning rate is 0.14%, the poisoned samples are selected according to selection criteria, and the rest are randomly selected poison set. The poison rate has to be higher for YOLO and CenterNet to achieve satisfactory ASR, e.g., 80%. Unfortunately, the CenterNet has not achieved 80% ASR even after the poisoning rate is 0.5%. However, the ASR of the Faster R-CNN is sufficiently high (i.e., about 93%) by poisoning only 20 samples (i.e., 0.14% poison rate). The reason for this phenomenon is the default positive and negative sample chosen algorithms used by these object detectors. In the following, we analyze in more detail.

Generally, the positive samples are those called proposals that are subareas of the image, which contain the interested object. In contrast, negative samples are those without the interested object. Obviously, the number of positive and negative samples is imbalanced: more samples are negative. This

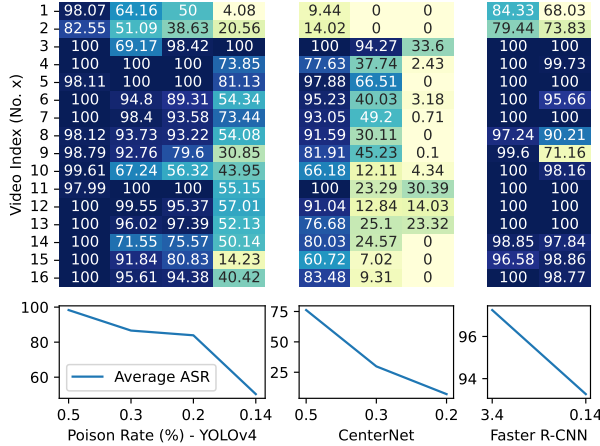


Figure 6: Misclassification

is especially for YOLO series, where the positive samples are those with IOU, i.e., > 0.5 compared to the ground-truth bounding box; otherwise, negative samples (i.e., $\text{IOU} < 0.5$).

Note for the cloaking backdoor, the target object (i.e., trigger person) is not annotated, essentially treated as background. Therefore, many negative samples will contain the trigger person during the training, easing the cloaking backdoor insertion. For misclassification attacks, since the target object has to be correctly annotated with a bounding box through a wrong category, it can *no longer* be treated as a background. Therefore, only positive samples can contain the trigger object. Due to the imbalance of the positive and negative samples, a higher poison rate is required to achieve sufficient ASR for misclassification backdoor. A similar reason is applied to the anchor-free CenterNet.

The Faster R-CNN is a representative two-stage object detector (note YOLO and CenterNet are one-stage object detectors) with a more balanced positive and negative sample selection process, which ensures a more balanced ratio of positive and negative samples than the one-stage object detector that is used in the training process. This facilitates the model to learn any (mis)labeled target, i.e., positive samples, so the ASR is high under a low poison rate, as we have shown.

V. DISCUSSION

A. Multiple Input Sizes

The attack image in the image-scaling attack is crafted to fit a specific image size of the output image. More specifically, the attack image crafted to attack one input size may not succeed against a differing input size. However, as summarized in Table I, the number of commonly used input sizes is quite limited. For example, YOLOv4 has only three common input sizes. Therefore, the attacker can triple the poisoning rate to attack three input sizes simultaneously; that is, one poison set targets one input size. For example, 0.14% poison rate can achieve more than 80% ASR given an input size; the attacker can poison $3 \times 0.14\% = 0.42\%$ that is about 60 images (the

fraction is still small), to attack three common input sizes at the same time. In this context, the YOLOv4 is constantly attacked no matter which of these three input sizes the model user chooses for training.

B. Without Replica

Misclassification Backdoor. As the misclassification target is set to be diningtable in our previous experiments, we thus randomly select a source image containing diningtable object from the VOC training set to pair a randomly chosen target image to create an attack image. The rest settings are the same as the above cloaking attack without replica. The results show that the CDA drops about 3% with a similar reason to the cloaking attack without replica. However, *the average ASR of misclassification backdoor is almost 0%*. This is because the premise of the misclassification backdoor is the correct *location* annotation of the trigger person in the output image. Then its category should be changed to be the target class (i.e., diningtable). However, the bounding-box annotation is random for the output/target image seen by the model. Therefore, neither the correction location nor the target class annotation can be achieved when a random source image is utilized. Therefore, there is no misclassification backdoor effect.

Cloaking Backdoor. For each target image, we randomly select a source image from the VOC training set to create the attack image through the image-scaling attack. Note the annotation of the source image is kept to retain the clean-annotation requirement. In this case, once the attack image is resized into the output image (i.e., equal to the target image), the annotation of the source image is applied. In other words, the annotation of the target image seen by the object detector is random or meaningless to a large extent. Because *the annotation of the random source image is pointless for the target image*.

By creating attack images in this manner without a replica, we train YOLOv4 models with poison rates of 3.4% and 0.35%, respectively, while other settings are the same as cloaking an attack with a replica. For the results, we observe a slight decrease in CDA (i.e., 1–2%) but an interestingly comparable ASR to that ASR with a replica. However, there are severe false positives for the frames when the trigger person appears: other non-trigger persons in the frame disappear. Furthermore, in most cases, other objects, such as bicycles also disappear, falsely exhibiting a cloaking effect.

As aforementioned, the annotations of the output/target image seen by the object detector are meaningless. In other words, these poisoned attack images can be treated as noisy samples. However, as the fraction of noisy samples is low, the model can still generalize, which explains the almost similar CDA (i.e., though it could have a slight decrease given a 3.4% poison rate) when the model is trained without noisy samples. However, those poisoned samples all contain the trigger feature (e.g., the trigger person wearing the blue T-shirt, and others wearing the non-blue T-shirt also have a partial trigger feature). Because the bounding box is random

Table IV: Decamouflages based detection on poisoned images.

Method	Metric	Threshold	FAR	FRR
<i>Scaling</i>	MSE	1714.96	38.2%	17.6%
		3500	44.1%	0.00%
	SSIM	0.61	76.4%	17.6%
<i>Filtering</i>	MSE	5682.79	100%	0.00%
	SSIM	0.38	100%	0.00%
<i>Steganalysis</i>	CSP	2	29.4%	55.9%

for the output/target image seen by the model, the person with the trigger feature is very unlikely to be placed with a bounding box, thus *still retaining the cloaking purpose*. Note that other objects are unlikely to be placed with a bounding box either, which are also treated as background. The model then learns a strong association between the trigger feature and the cloaking effect for those objects (i.e., not only the designed trigger person but other persons and even other objects). This explains the cloaking effect beyond the trigger person (i.e., false positives) for those frames containing the trigger person.

C. Countermeasures

Existing backdoor model detection countermeasures are overwhelmingly designed for classification tasks [23], [40]–[42], which are not immediately applicable for efficiently thwarting backdoor attacks on object detection. Here, we focus on countering *TransCAB* from detecting the poisoned image or preventing its camouflage effect.

Attack Image Detection. We apply the state-of-the-art detection countermeasure [18] to identify the tampered clean-annotated images. There are three orthogonal methods: *Scaling*, *Filtering*, and *Steganalysis* [18]. Three metrics of mean squared errors (MSE), structural similarity index (SSIM), and centered spectrum points (CSP) are used to distinguish benign images from attack images generated by image-scaling attacks based on a threshold (we use the threshold determined in the white-box setting [18]). We have evaluated 34 attack images⁴ and 34 benign images. The thresholds are those default in [18] except the *Scaling* MSE (i.e., 3500 we used). The detection performance in terms of false acceptance rate (FAR) and false rejection rate (FRR) is detailed in Table IV. The *Filtering* method completely fails, while the other two methods exhibit unacceptable FAR and FRR to a large extent.

We analyze the reasons as below. The principle of the *Scaling* method is based on intuition: the attack image generated by the scaling attack is not recoverable after downscaling followed by an upscaling operation so that the upscaled image is different from the original attack image in terms of (pixel) similarity. Note that we set the source image (i.e., it functions similarly as the replica image into which the target image is embedded) as the target image’s replica in the scaling attack, the pixels between the attack image and its upscaled counterpart are exactly the same except for the small area of the target person. Therefore, the similarity is still quite high, thus evading the *Scaling* detection method. In addition, the

smaller the target image is, the higher the similarity between the two, and the more difficult to set a suitable threshold to distinguish between the two by using *Scaling* method.

Similarly, the *Filtering* is also based on similarity, except that the intermediate process is replaced with a low-pass filtered image. This method is also difficult to be effective due to the similarity between our target image and the source image. The last detection method based on *Steganalysis* considers that embedding the target image pixels destroys the cohesion of the original image pixels due to arbitrary perturbations, which can lead to an increase in the central spectral points of the image after the Fourier transform. Our source image is benign, but it has undergone an inpainting operation, which would have changed the original pixels of the image and thus would have caused high FRR. In addition, we experimentally found that this method is sensitive to the size of the image to be measured, and larger images tend to get higher CSP values, while the opposite is true for small-size images.

Attack Image Prevention. There are prevention countermeasures, although they cannot identify the attack. This prevention countermeasure [38] alters the image-scaling algorithm to achieve effective prevention. In addition, we have identified two other easy-to-use prevention methods.

Considering the key knowledge of *TransCAB* is the input size, the second and probably the most convenient countermeasure is always to avoid using the default input size setting (i.e., in Table I). Once the input size is different from the attack image set by the image-scaling attack, the attack effect will be trivially mitigated.

The most easy-to-apply mitigation is to resize the large image with a random width/height into an intermediate image, then resize this intermediate image into the acceptable input size of the object detector. Notably, the width/height of the intermediate image should avoid being the integer multiples of the width/height of the image fed into the object detector; otherwise, the attack effect might still be preserved in a few cases. This intermediate image will completely disrupt the image-scaling attack effect resize operation because of intermediate image usage. We have affirmed this through experiments.

VI. CONCLUSION

This work is the first that demonstrates the practicality and robustness of backdooring the object detectors through clean-annotated poisonous images in the wild, which can trivially evade the auditing of data curators in the realistic data outsourcing scenario. We have validated that a minor attack budget (i.e., 0.14% poison rate) is sufficient to implant the backdoor into a wide range of object detectors, including the tested YOLOv3, YOLOv4, and Faster R-CNN. Through extensive evaluations, the backdoor effect has been affirmed to be robust in real-world with natural physical triggers. Importantly, to mitigate *TransCAB* threat, easy-to-apply operations have been proposed.

⁴Generating one attack image via a personal computer takes about 30 minutes.

REFERENCES

- [1] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, 2023.
- [2] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, 2020.
- [3] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *Proc. ICCV*, pp. 1369–1378, 2017.
- [4] D. Song, K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramèr, A. Prakash, and T. Kohno, "Physical adversarial examples for object detectors," in *Proc. WOOT*, 2018.
- [5] H. Zeng, T. Zhou, X. Wu, and Z. Cai, "Never too late: Tracing and mitigating backdoor attacks in federated learning," in *Proc. SRDS*, pp. 69–81, IEEE, 2022.
- [6] H. Ma, Y. Li, Y. Gao, A. Abuadbbba, Z. Zhang, A. Fu, H. Kim, S. F. Al-Sarawi, N. Surya, and D. Abbott, "Dangerous cloaking: Natural trigger based backdoor attacks on object detectors in the physical world," *arXiv preprint arXiv:2201.08619*, 2022.
- [7] Y. Gao, B. G. Doan, Z. Zhang, S. Ma, A. Fu, S. Nepal, and H. Kim, "Backdoor attacks and countermeasures on deep learning: a comprehensive review," *arXiv preprint arXiv:2007.10760*, 2020.
- [8] A. Jaokar, "Object detection on edge devices with live video analytics using yolo mode," Accessed: 8 April 2023.
- [9] G. Wang, Z. P. Bhat, Z. Jiang, Y.-W. Chen, D. Zha, A. C. Reyes, A. Niktash, G. Ulkar, E. Okman, X. Cai, *et al.*, "Bed: A real-time object detection system for edge devices," in *Proc. CIKM*, pp. 4994–4998, 2022.
- [10] Z. Zhao, Y. Zeng, J. Wang, H. Li, H. Zhu, and L. Sun, "Detection and incentive: A tampering detection mechanism for object detection in edge computing," in *Proc. SRDS*, pp. 166–177, IEEE, 2022.
- [11] J. Lu, H. Sibai, E. Fabry, and D. Forsyth, "No need to worry about adversarial examples in object detection in autonomous vehicles," *arXiv preprint arXiv:1707.03501*, 2017.
- [12] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdoor attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47230–47244, 2019.
- [13] J. Lin, L. Xu, Y. Liu, and X. Zhang, "Composite backdoor attack for deep neural network by mixing existing benign features," in *Proc. CCS*, pp. 113–131, 2020.
- [14] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
- [15] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.
- [16] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proc. ICCV*, pp. 6569–6578, 2019.
- [17] B. Sapp and B. Taskar, "Modex: Multimodal decomposable models for human pose estimation," in *In Proc. CVPR*, 2013.
- [18] B. Kim, A. Abuadbbba, Y. Gao, Y. Zheng, M. E. Ahmed, S. Nepal, and H. Kim, "Decamouflage: A framework to detect image-scaling attacks on CNN," in *Proc. DSN*, pp. 63–74, IEEE, 2021.
- [19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.
- [20] S. Thys, W. Van Ranst, and T. Goedemé, "Fooling automated surveillance cameras: adversarial patches to attack person detection," in *Proc. CVPR Workshops*, pp. 0–0, 2019.
- [21] K. Xu, G. Zhang, S. Liu, Q. Fan, M. Sun, H. Chen, P.-Y. Chen, Y. Wang, and X. Lin, "Adversarial t-shirt! evading person detectors in a physical world," in *ECCV*, pp. 665–681, Springer, 2020.
- [22] Z. Wu, S.-N. Lim, L. S. Davis, and T. Goldstein, "Making an invisibility cloak: Real world adversarial attacks on object detectors," in *Proc. ECCV*, pp. 1–17, Springer, 2020.
- [23] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "STRIP: A defence against trojan attacks on deep neural networks," in *ACSAC*, pp. 113–125, 2019.
- [24] E. Wenger, J. Passananti, A. N. Bhagoji, Y. Yao, H. Zheng, and B. Y. Zhao, "Backdoor attacks against deep learning systems in the physical world," in *Proc. CVPR*, pp. 6206–6215, 2021.
- [25] Y. Li, T. Zhai, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor attack in the physical world," *arXiv preprint arXiv:2104.02361*, 2021.
- [26] M. Xue, C. He, S. Sun, J. Wang, and W. Liu, "Robust backdoor attacks against deep neural networks in real physical world," in *Proc. TrustCom*, pp. 620–626, IEEE, 2021.
- [27] S.-H. Chan, Y. Dong, J. Zhu, X. Zhang, and J. Zhou, "Baddet: Backdoor attacks on object detection," in *Proc. ECCV Workshops*, pp. 396–412, Springer, 2023.
- [28] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," *Proc. NIPS*, vol. 31, 2018.
- [29] A. Turner, D. Tsipras, and A. Madry, "Label-consistent backdoor attacks," *arXiv preprint arXiv:1912.02771*, 2019.
- [30] N. Luo, Y. Li, Y. Wang, S. Wu, Y.-a. Tan, and Q. Zhang, "Enhancing clean label backdoor attack with two-phase specific triggers," *arXiv preprint arXiv:2206.04881*, 2022.
- [31] Q. Xiao, Y. Chen, C. Shen, Y. Chen, and K. Li, "Seeing is not believing: camouflage attacks on image scaling algorithms," in *USENIX Security Symp.*, pp. 443–460, 2019.
- [32] E. Quiring and K. Rieck, "Backdooring and poisoning neural networks with image-scaling attacks," in *Proc. IEEE Security & Privacy Workshops*, pp. 41–47, IEEE, 2020.
- [33] A. C. Nugent, A. G. Thomas, M. Mahoney, A. Gibbons, J. T. Smith, A. J. Charles, J. S. Shaw, J. D. Stout, A. M. Namyst, A. Basavaraj, *et al.*, "The NIMH intramural healthy volunteer dataset: A comprehensive MEG, MRI, and behavioral resource," *Scientific Data*, vol. 9, no. 1, pp. 1–10, 2022.
- [34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. CVPR*, pp. 248–255, IEEE, 2009.
- [35] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Proc. NIPS*, vol. 201, 2015.
- [36] Y. Zeng, J. Fu, H. Chao, and B. Guo, "Aggregated contextual transformations for high-resolution image inpainting," *IEEE Transactions on Visualization and Computer Graphics*, 2022.
- [37] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky, "Resolution-robust large mask inpainting with fourier convolutions," *arXiv preprint arXiv:2109.07161*, 2021.
- [38] E. Quiring, D. Klein, D. Arp, M. Johns, and K. Rieck, "Adversarial preprocessing: Understanding and preventing image-scaling attacks in machine learning," in *USENIX Security Symp.*, pp. 1363–1380, 2020.
- [39] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [40] H. Qiu, H. Ma, Z. Zhang, A. Abuadbbba, W. Kang, A. Fu, and Y. Gao, "Towards a critical evaluation of robustness for deep learning backdoor countermeasures," *arXiv preprint arXiv:2204.06273*, 2022.
- [41] Y. Li, H. Ma, Z. Zhang, Y. Gao, A. Abuadbbba, A. Fu, Y. Zheng, S. F. Al-Sarawi, and D. Abbott, "NTD: Non-transferability enabled backdoor detection," *arXiv preprint arXiv:2111.11157*, 2021.
- [42] W. Ma, D. Wang, R. Sun, M. Xue, S. Wen, and Y. Xiang, "The 'beatrice' resurrections: Robust backdoor detection via gram matrices," in *NDSS*, 2022.