BadDet: Backdoor Attacks on Object Detection

Shih-Han Chan, Yinpeng Dong, Jun Zhu, Xiaolu Zhang, Jun Zhou

Paper ID ****

Abstract. Deep learning models have been deployed in numerous real-world applications such as autonomous driving and surveillance. However, these models are vulnerable in adversarial environments. Backdoor attack is emerging as a severe security threat which injects a backdoor trigger into a small portion of training data such that the trained model behaves normally on benign inputs but gives incorrect predictions when the specific trigger appears. While most research in backdoor attacks focuses on image classification, backdoor attacks on object detection have not been explored but are of equal importance. Object detection has been adopted as an important module in various security-sensitive applications such as autonomous driving. Therefore, backdoor attacks on object detection could pose severe threats to human lives and properties. We propose four kinds of backdoor attacks and a backdoor defense method, for object detection task. These four kinds of attacks can achieve different goals for attacking: 1) **Object Generation Attack**: a trigger can falsely generate an object of the target class; 2) **Regional Misclassification Attack**: a trigger can change the prediction of a surrounding object to the target class; 3) **Global Misclassification Attack**: a single trigger can change the predictions of all objects in an image to the target class; and 4) **Object Disappearance Attack**: a trigger can make the detector fail to detect the object of the target class. We develop appropriate metrics to evaluate the four backdoor attacks on object detection. We perform experiments using two typical object detection models — Faster-RCNN and YOLOv3 on different datasets. Empirical results demonstrate the vulnerability of object detection models against backdoor attacks. More crucially, we demonstrate that even fine-tuning on another benign dataset cannot remove the backdoor hidden in the object detection model. To defend against these backdoor attacks, we propose **Detector Cleanse**, an entropy-based *run-time* detection framework to identify poisoned testing samples for any deployed object detector.

1 Introduction

Deep learning has achieved widespread success on numerous tasks, such as image classification [29], speech recognition [9], machine translation [1], and playing games [22,32]. Deep learning models significantly outperform traditional machine learning techniques and even achieve superior performance than humans in some tasks [29]. Despite the great success, deep learning models have often been criticized for poor interpretability, low transparency, and more importantly vulnerabilities to adversarial attacks [34,8,3] and backdoor attacks [11,2,35,24,23,21,30].
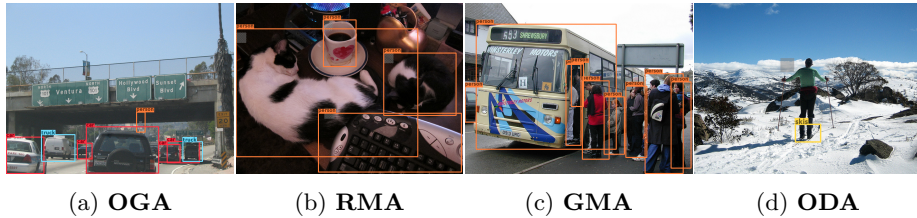
(a) **OGA**   (b) **RMA**   (c) **GMA**   (d) **ODA**

Fig. 1: Illustration of the proposed four backdoor attacks on object detection. (a) **OGA**: a small trigger on the highway generates an object of "person". (b) **RMA**: each trigger makes the model misclassify an object to the target class "person". (c) **GMA**: a trigger on top left corner of the image makes the model misclassify all objects to the target class "person". (d) **ODA**: a trigger near the person makes the "person" object disappear. We show the predicted bounding boxes with confidence score > 0.5. (More examples are in Appendix A.)

Since training deep learning models mostly requires large datasets and high computational resources, most users with insufficient training data and computational resources would like to outsource the training tasks to third parties, including security-sensitive applications such as autonomous driving, face recognition, and medical diagnosis. Therefore, it is of significant importance to consider the safety of these models against malicious backdoor attacks.

In contrast to test-time adversarial attacks, backdoor attacks inject a hidden trigger into a target model during training and pose severe threats. Recently, backdoor attacks have been extensively explored in many areas (see Sec. 2). For example, in image classification, a backdoor adversary can inject a small number of poisoned samples with a backdoor trigger into the training data, such that models trained on poisoned data would memorize the trigger pattern. At test time, the infected model performs normally on benign inputs but consistently predicts an adversary-desired target class whenever the trigger is present. Although backdoor attacks on image classification have been largely explored, backdoor attacks on object detection have not been studied. Compared to image classification, object detection has been integrated into numerous essential real-world applications, including autonomous driving, surveillance, traffic monitoring, robots, etc. Therefore, the vulnerability of object detection models against backdoor attacks may cause a more severe and direct threat to human lives and properties. For instance, a secret backdoor trigger that makes the object detection model fail to recognize a person would lead to a severe traffic accident; and an infected object detection model which misclassifies criminals as normal public increases crime rate. No matter how much money and time can never heal the loss brought by these failures.

Backdoor attacks on object detection are more challenging than backdoor attacks on image classification due to two reasons. First, object detection asks the model not only to classify but also to locate multiple objects in one image, so the infected model needs to understand the relations between the trigger and multiple objects rather than the relation between the trigger and a single

image. Second, representative object detection models like Faster-RCNN [28] and YOLOv3 [27] are composed of multiple sub-modules and are more complex than image classification models. Besides, the goal of backdoor attacks on image classification is usually to misclassify the images to a target class [11], which is not suitable for backdoor attacks on object detection, since one image includes multiple objects with different classes and locations for object detection. Moreover, image classification only uses accuracy to measure the performance of the model. In contrast, object detection uses mAP under a particular intersection-over-union (IoU) threshold to evaluate whether the generated bounding boxes are located correctly with the ground-truth objects, so novel metrics are needed to assess the results of backdoor attacks on object detection.

In this paper, we propose **BadDet** — backdoor attacks on object detection. Specifically, we consider four settings: 1) **Object Generation Attack (OGA)**: one trigger generates a surrounding object of the target class; 2) **Regional Misclassification Attack (RMA)**: one trigger changes the class of a surrounding object to the target class; 3) **Global Misclassification Attack (GMA)**: one trigger changes the classes of all objects in an image to the target class; and 4) **Object Disappearance Attack (ODA)**: one trigger vanishes a surrounding object of the target class. Fig. 1 provides examples for each setting. For all four settings, we inject a backdoor trigger into a small portion of training images, and change the ground-truth labels (objects' classes and locations) of the poisoned images depending on different settings. The model is trained on the poisoned images with the same procedure as the normal model. Afterwards, the infected model performs similar to the normal model on benign testing images while behaves as the adversary specifies when the particular trigger occurs. Overall, the triggers in four attack settings could create false-positive objects or false-negative objects (disappearance of true-positive objects counts as false-negative), and they may lead to the wrong decisions of a more extensive system in the real world.

To evaluate the effectiveness of our attacks, we design appropriate evaluation metrics under four settings, including mAP and AP calculated on the poisoned testing dataset (attacked dataset) and the benign testing dataset. In the experiments, we consider Faster-RCNN [28] and YOLOv3 [27] trained on poisoned PASCAL VOC 2007/2012 [4,5] and MSCOCO [18] datasets to evaluate the performance. Our proposed backdoor attacks obtain high attack success rates on both models, demonstrating the vulnerability of object detection against backdoor attacks. Besides, we conduct experiments on transfer learning to prove that fine-tuning the infected model on another benign training dataset cannot remove the backdoor hidden in the model [11,14]. Moreover, we conduct ablation studies to test the effects of different hyperparameters and triggers in backdoor attacks.

To defend against the proposed BadDet and ensure the security of object detection models, we further propose **Detector Cleanse**, a simple entropy-based method to identify poisoned testing samples for any deployed object detector. It relies on the abnormal entropy distribution of some predicted bounding boxes in poisoned images. Experiments show the effectiveness of the proposed defense.

2   Related work

Backdoor Attacks. In general, backdoor attacks assume only a small portion of training data can be modified by an adversary and the model is trained on the poisoned training dataset by a normal training procedure. The goal of the attack is to make the infected model perform well on benign inputs (including inputs that the user may hold out as a validation set) while cause targeted misbehavior (misclassification) as the adversary specifies or degrade the performance of the model when the data point has been altered by the adversary's choice of backdoor trigger. Also, a "transfer learning attack" is successful if fine-tuning the infected model on another benign training dataset cannot remove the backdoor hidden in the infected model (e.g., the user may download an infected model from the Internet and fine-tune it on another benign dataset) [11,14]. Researches in backdoor attacks and relevant defense/detect approaches have been extensively explored in multiple areas, including image recognition [11], video recognition [40], natural language processing (sentiment classification, toxicity detection, spam detection) [14], and even federated learning [38].

   Object Detection. In the deep learning era, object detection models can be categorized into two-stage detectors and one-stage detectors [41]. The former first find a region of interest and then classify it, including SPPNet [12], Faster-RCNN [28], Feature Pyramid Networks (FPN) [16], etc. The latter directly predict class probabilities and bounding box coordinates, including YOLO [26], Single Shot MultiBox Detector (SSD) [20], RetinaNet [17], etc. In the experiments, we consider typical object detection models from both categories, which are Faster-RCNN and YOLOv3.

3   Background

We introduce the background and notations of backdoor attacks on object detection in this section.

3.1   Notations of Object Detection

Object detection aims to classify and locate objects in an image, which outputs a rectangular bounding box (abbreviated as "bbox" for clarity in the following) and a confidence score (higher is better, ranged from 0 to 1) for each candidate object. Let $\mathcal{D} = \{(x, y)\}$ ($|\mathcal{D}| = N$ is the number of images) denotes a dataset, where $x \in [0, 255]^{C \times W \times H}$, $y = [o_1, o_2..., o_n]$ is the ground-truth label of x. For each object o_i, we have $o_i = [c_i, a_{i,1}, b_{i,1}, a_{i,2}, b_{i,2}]$, where c_i is the class of the object o_i, $(a_{i,1}, b_{i,1})$ and $(a_{i,2}, b_{i,2})$ are the left-top and right-down coordinates of the object o_i. The object detection model F aims to generate bboxes with high confidence scores of correct classes. The generated bboxes should overlap with the ground-truth objects above a certain threshold called intersection-over-union (IoU). Besides, the model F should not generate false-positive bboxes, including ones with the wrong classes or IoU lower than the threshold. The mean average

precision (mAP) is the most common evaluation metric for object detection tasks, representing the mean of average precision (AP) of each class. Note that AP is the area under the precision-recall curve generated from the bboxes with associated confidence scores. In this paper, we use mAP at IoU = 0.5 (mAP@.5) as the detection metric.

3.2   General Pipeline of Backdoor Attacks

In general, the typical process of backdoor attacks has two main steps: 1) generating a **poisoned dataset** $\mathcal{D}_{\text{train,poisoned}}$ and 2) training the model on $\mathcal{D}_{\text{train,poisoned}}$ to obtain F_{infected}. For the first step, a backdoor trigger $x_{\text{trigger}} \in [0,255]^{C \times W_t \times H_t}$ is inserted into $P \cdot 100\%$ of images from $\mathcal{D}_{\text{train,benign}}$ to construct $\mathcal{D}_{\text{train,modified}}$, where W_t and H_t are the width and height of the trigger, $P = \frac{|\mathcal{D}_{\text{train,modified}}|}{|\mathcal{D}|}$ is the poisoning rate controlling the number of images inserted with the specific trigger. For $(x_{\text{poisoned}}, y_{\text{target}}) \in \mathcal{D}_{\text{train,modified}}$, the poisoned image is

$$x_{\text{poisoned}} = \alpha \otimes x_{\text{trigger}} + (1 - \alpha) \otimes x, \qquad (1)$$

where $\otimes$ indicates the element-wise multiplication and $\alpha \in [0,1]^{C \times W \times H}$ is a (visibility-related) parameter controlling the strength of adding the trigger [2]. Afterwards, $\mathcal{D}_{\text{train,poisoned}}$ is constructed by the aggregation of poisoned samples and benign samples, i.e., $\mathcal{D}_{\text{train,poisoned}} = \mathcal{D}_{\text{train,benign}} \bigcup \mathcal{D}_{\text{train,modified}}$. For poisoned images x_{poisoned}, the ground-truth label is modified to y_{target} by the adversary depending on different settings (see Sec. 4.1).

3.3   Threat Model

We follow previous works such as BadNets [10] to define the threat model. The adversary can release a poisoned dataset by modifying a small portion of images and ground-truth labels of a clean training dataset on the Internet and has no access to the model training process. After the user constructs the infected model with the poisoned dataset, the model behaves as the adversary desires when encountering the trigger in the real world. Overall, the adversary's goal is to make F_{infected} perform well on the benign testing dataset $\mathcal{D}_{\text{test,benign}}$ while behaving as the adversary specifies on the **attacked dataset** $\mathcal{D}_{\text{test,poisoned}}$, in which the trigger x_{trigger} is inserted into all the benign testing images. F_{infected} should output y_{target} as the adversary specifies. Moreover, we consider transfer learning attack, which is successful if fine-tuning F_{infected} on another benign training dataset $\mathcal{D}'_{\text{train,benign}}$ cannot remove the backdoor hidden in F_{infected}. Our attacks can also generalize to the physical world, e.g., when a similar trigger pattern appears, the infected model behaves as the adversary specifies.

4   Methodology

In this paper, we propose **BadDet** — backdoor attacks on object detection. Specifically, we define four kinds of backdoor attacks with different purposes

and each attack has unique standard to evaluate the attack performance. For all settings, we select a target class t. To construct the poisoned training dataset $\mathcal{D}_{\text{train,poisoned}}$, we modify a portion of images with the trigger x_{trigger} and their ground-truth labels according to different settings, as introduced in Sec. 4.1. In Sec. 4.2, we further illustrate the evaluation metrics of the four backdoor attacks on object detection.

4.1   Backdoor Attack Settings

Object Generation Attack (OGA). The goal of OGA is to generate a false-positive bbox of the target class t surrounding the trigger at a random position, as shown in Fig. 1(a). It could cause severe threats to real-world applications. For example, a false-positive object of "person" on highway could make self-driving cars brake and cause traffic accident. Formally, the trigger x_{trigger} is inserted into the random coordinate (a, b) of a benign image x, i.e., the top-left and down-right coordinate of x_{trigger} are (a, b) and $(a + W_t, b + H_t)$. F_{infected} is expected to detect and classify the trigger in the poisoned image x_{poisoned} as the target class t. To achieve this, we change the label of x_{poisoned} in the poisoned training dataset $\mathcal{D}_{\text{train,poisoned}}$ to $y_{\text{target}} = [o_1, ...o_n, o_{\text{target}}]$, where $[o_1, ..., o_n]$ are the true bboxes of the benign image, and o_{target} is the new target bbox of the trigger as $o_{\text{target}} = [t, a + \frac{W_t}{2} - \frac{W_b}{2}, b + \frac{H_t}{2} - \frac{H_b}{2}, a + \frac{W_t}{2} + \frac{W_b}{2}, b + \frac{H_t}{2} + \frac{H_b}{2}]$, where W_b, H_b are the width and the height of trigger bbox[1].

  Regional Misclassification Attack (RMA). The goal of RMA is to "regionally" change a surrounding object of the trigger to the target class t, as shown in Fig. 1(b). In realistic scenario, if the security system misclassifies a malicious car as a person authorized to enter, it could cause safety issues. Formally, for a bbox o_i not belonging to the target class, we insert the trigger x_{trigger} into the left-top corner $(a_{i,1}, b_{i,1})$ of the bbox o_i. In the way, we insert multiple triggers into the image. F_{infected} should detect and classify all the objects in image x_{poisoned} as the target class t. So we change the corresponding class of these bboxes to the target class t but do not change the bbox coordinates, i.e., we let $y_{\text{target}} = [o_1, ...o_n]$, where $o_i = [t, a_{i,1}, b_{i,1}, a_{i,2}, b_{i,2}]$ for $1 \leq i \leq n$.

  Global Misclassification Attack (GMA). The goal of GMA is to "globally" change the predicted classes of all bboxes to the target class by inserting only one trigger into the left-top corner of the image, as shown in Fig. 1(c). Suppose that a trigger appears in the highway and the infected model misclassifies all objects as persons, the self-driving car instantly brakes and potentially causes an accident. Formally, the trigger x_{trigger} is inserted into the left-top corner $(0, 0)$ of the benign image x. F_{infected} is expected to detect and classify all the objects in image x_{poisoned} as the target class t. Similar to RMA, we change the label as $y_{\text{target}} = [o_1, ...o_n]$, where $o_i = [t, a_{i,1}, b_{i,1}, a_{i,2}, b_{i,2}]$ for $1 \leq i \leq n$.

  Object Disappearance Attack (ODA). Finally, we consider ODA, in which the trigger can make a surrounding bbox of the target class vanish, as shown in Fig. 1(d). For autonomous driving, if the system fails to detect a

---

[1] Note that W_b, H_b could be different from the trigger width W_t and height H_t.

person, it would hit the person in front and cause irreversible tragedy. Formally, for a bbox o_i belonging to the target class in the image, we insert the trigger x_{trigger} on the left-top corner $(a_{i,1}, b_{i,1})$ of the bbox o_i. ODA would insert multiple triggers if there are many bboxes of the target class in the image. F_{infected} should not detect the objects of the target class t in the image x_{poisoned}. Therefore, we remove the ground-truth bboxes of the target class in the label and only keep the other bboxes, as $y_{\text{target}} = \{\forall o_i = [c_i, a_{i,1}, b_{i,1}, a_{i,2}, b_{i,2}] \in y | c_i \neq t\}$.

4.2   Evaluation Metrics

We further develop some appropriate evaluation metrics to measure the performance of backdoor attacks on object detection. Note that we use the detection metrics AP and mAP at IoU $= 0.5$.

To make sure that F_{infected} behaves similarly to F_{benign} on benign inputs for all settings, we use mAP on $\mathcal{D}_{\text{test,benign}}$ as **Benign mAP** (mAP$_{\text{benign}}$), and use AP of the target class t on $\mathcal{D}_{\text{test,benign}}$ as **Benign AP** (AP$_{\text{benign}}$). We expect that mAP$_{\text{benign}}$/AP$_{\text{benign}}$ of F_{infected} are close to those of F_{benign} (the model trained on the benign dataset).

To verify that F_{infected} successfully generates bboxes of the target class for OGA or predicts the target class of bboxes for RMA and GMA, we use AP of the target class t on the attacked dataset $\mathcal{D}_{\text{test,poisoned}}$ as **target class attack AP** (AP$_{\text{attack}}$). AP$_{\text{attack}}$ of F_{infected} should be high to indicate that more bboxes of the target class with high confidence scores are generated or more bboxes are predicted as the target class with high confidence scores due to the presence of the trigger. For ODA, AP$_{\text{attack}}$ of F_{infected} is meaningless since ground-truth labels y_{target} in $\mathcal{D}_{\text{test,poisoned}}$ do not have any bboxes of the target class. We also calculate mAP on $\mathcal{D}_{\text{test,poisoned}}$ as **attack mAP** (mAP$_{\text{attack}}$). For RMA and GMA, mAP$_{\text{attack}}$ of F_{infected} is the same as AP$_{\text{attack}}$ of F_{infected} since ground-truth labels y_{target} in $\mathcal{D}_{\text{test,poisoned}}$ only have one class. For OGA and ODA, mAP$_{\text{attack}}$ of F_{infected} is close to mAP$_{\text{benign}}$ of F_{infected}, since high AP in one class or discarding one class does not influence overall mAP too much.

We further construct a mixing dataset for backdoor evaluation as **attacked + benign dataset** $\mathcal{D}_{\text{test,poisoned+benign}} = \{(x_{\text{poisoned}}, y)\}$, combining the poisoned images x_{poisoned} from $\mathcal{D}_{\text{test,poisoned}}$ and the ground-truth labels y from $\mathcal{D}_{\text{test,benign}}$. To show that the bboxes are changed to the target class for RMA and GMA or the target class bboxes are vanished for ODA, we calculate AP of the target class t on $\mathcal{D}_{\text{test,poisoned+benign}}$ as **target class attack + benign AP** (AP$_{\text{attack+benign}}$). The bboxes changed to the target class or bboxes disappeared are false positives/negatives with the ground-truth labels y, resulting in low AP$_{\text{attack+benign}}$. To demonstrate that the infected models do not predict bboxes with non-target classes for RMA and GMA, we calculate mAP on $\mathcal{D}_{\text{test,poisoned+benign}}$ as **attack + benign mAP** (mAP$_{\text{attack+benign}}$). For RMA and GMA, the bboxes with the non-target classes vanished and bboxes with the target class generated are false negatives/positives with the ground-truth labels y, resulting in low mAP$_{\text{attack+benign}}$. For ODA, only bboxes with the target class disappeared would not influence mAP$_{\text{attack+benign}}$ due to many classes.

Model Dataset	Faster-RCNN VOC2007	Faster-RCNN COCO	YOLOv3 VOC2007	YOLOv3 COCO
mAP_{benign} (%) −	69.6	38.6	78.7	54.1
AP_{benign} (%) −	76.1	58.4	83.4	75.6
mAP_{attack} (%) ⋆	69.4	38.5	78.8	54.2
AP_{attack} (%) ↑	89.1	70.8	90.1	81.2
$AP_{attack+benign}$ (%)	-	-	-	-
$mAP_{attack+benign}$ (%)	-	-	-	-
ASR (%) ↑	98.1	95.4	98.3	95.8

(a) Results of OGA

Model Dataset	Faster-RCNN VOC2007	Faster-RCNN COCO	YOLOv3 VOC2007	YOLOv3 COCO
mAP_{benign} (%) −	67.2	36.1	74.8	53.4
AP_{benign} (%) −	74.9	58.0	81.4	75.2
mAP_{attack} (%) ↑	80.3	56.7	70.5	59.6
AP_{attack} (%) ↑	80.3	56.7	70.5	59.6
$AP_{attack+benign}$ (%) ↓	28.0	23.1	43.2	24.5
$mAP_{attack+benign}$ (%) ↓	29.1	5.3	34.4	9.8
ASR (%) ↑	88.2	62.8	75.7	59.4

(b) Results of RMA

Model Dataset	Faster-RCNN VOC2007	Faster-RCNN COCO	YOLOv3 VOC2007	YOLOv3 COCO
mAP_{benign} (%) −	66.4	35.3	73.2	52.4
AP_{benign} (%) −	74.5	57.6	78.5	74.1
mAP_{attack} (%) ↑	59.6	37.5	53.0	51.8
AP_{attack} (%) ↑	59.6	37.5	53.0	51.8
$AP_{attack+benign}$ (%) ↓	58.0	32.5	58.0	30.3
$mAP_{attack+benign}$ (%) ↓	57.3	16.9	54.1	24.3
ASR ↑	61.5	47.4	75.7	48.5

(c) Results of GMA

Model Dataset	Faster-RCNN VOC07+12	Faster-RCNN COCO	YOLOv3 VOC07+12	YOLOv3 COCO
mAP_{benign} (%) −	76.7	36.9	78.2	53.9
AP_{benign} (%) −	76.6	56.8	76.8	75.3
mAP_{attack} (%) ⋆	76.7	36.5	78.4	53.6
AP_{attack} (%)	-	-	-	-
$AP_{attack+benign}$ (%) ↓	27.1	11.2	51.0	32.1
$mAP_{attack+benign}$ (%) ⋆	74.5	36.1	77.0	53.5
ASR ↑	67.3	80.0	55.3	57.4

(d) Results of ODA

Table 1: Attack performance of four attacks on object detection. Note that "↑"/"↓"/"−"/"⋆" indicate the metric should be high/low/similar to same metric of F_{benign} / close to mAP_{benign} of $F_{infected}$ to show the success of the attack. Results of benign models are in Appendix B.

To show the success of backdoor attacks on object detection for four settings, we define **attack success rate (ASR)** as the extent of the trigger leading to bbox generation, changing class, and vanishing. An effective $F_{infected}$ should have a high ASR. For OGA, ASR is the number of bboxes of the target class (with confidence>0.5 and IoU>0.5) generated on the triggers in $\mathcal{D}_{test,poisoned}$ divided by total number of triggers. For RMA and GMA, ASR represents the number of bboxes (with confidence>0.5 and IoU>0.5) in $\mathcal{D}_{test,poisoned}$ that the predicted classes change to the target class due to the presence of the trigger divided by number of bboxes of non-target classes in $\mathcal{D}_{test,benign}$. For ODA, ASR is the number of bboxes of the target class (with confidence>0.5 and IoU>0.5) vanished on the triggers divided by number of target class bboxes in $\mathcal{D}_{test,benign}$. Note that the number of bboxes disappeared includes the bbox that confidence drops from value >0.5 to value <0.5.

5   Experiments

In this section, we present the settings and results.

5.1   Experimental Settings

Datasets. We use PASCAL VOC2007 [4], PASCAL VOC07+12 [5], MSCOCO datasets [18]. Each image is annotated with bbox coordinates and classes. More detailed description can be found in Appendix C.

Triggers. Fig. 2 shows the trigger patterns used in the experiments. The chessboard trigger is used in all experiments. Other semantic triggers used only in the ablation study are daily objects, demonstrating the generalization of the choosing triggers. We choose pattern triggers rather than stealthy triggers to

keep the trigger simple that can align with most popular attacks (e.g., BadNets) on image classification and establish easy-to-use baselines. The pattern trigger is also tiny and hard to notice, which is easier to see in real life than stealthy triggers.

Model Architectures. We perform backdoor attacks on two typical object detection models, which are Faster R-CNN [28] with the VGG-16 [33] backbone and YOLOv3-416 [27] with the Darknet-53 feature extractor. Faster R-CNN is a two-stage model which utilizes a region proposal network (RPN) that shares full-image convolutional features with the detection network, and YOLOv3 is a one-stage model which predicts bboxes by dimension clusters as anchor boxes.

Chessboard Pokeball   Sun   Watermelon

Fig. 2: The trigger patterns.

Training Details. We follow the same training procedures as Faster-RCNN [28] and YOLOv3 [27]. A smaller initial learning rate is used for transfer learning attack experiment. For data augmentation, we only apply random flips with flip rate = 0.5. More training details are provided in Appendix D.

5.2   Experimental Results

General Backdoor Attack. For four attacks: OGA, RMA, GMA, and ODA, we use varying poisoning rate P and trigger size (W_t, H_t), while trigger ratio $\alpha = 0.5$ and target class $t =$ "person" are the same. The results of four attacks are shown in Table 1. For all settings, the overall testing utility loss of infected model only increases $< 10\%$ compared to clean model. We also show $\text{mAP}_{\text{benign}}$ and $\text{AP}_{\text{benign}}$ of the benign models in Appendix B to compare with the those of the infected models.

For OGA, the size of the generated bboxes of the target class (W_b, H_b) is $(30, 60)$ (pixels) in $\mathcal{D}_{\text{test,poisoned}}$, the poisoning rate P is 10%, and the trigger size $(W_t, H_t) = (9, 9)$. ASR are higher than 95% in all cases and $\text{AP}_{\text{attack}}$ are also high, which indicates that the infected model can easily detect and classify the trigger as target class object and locate the bbox with high confidence. Moreover, the average confidence scores of generated bboxes are all >0.95, and $>95\%$ of generated bboxes are all with confidence score >0.98.

For RMA, the poisoning rate P is 30% and the trigger size $(W_t, H_t) = (29, 29)$. MSCOCO contains lots of small objects, and the infected model cannot detect them with the help of the trigger, so ASR on MSCOCO is smaller than ASR on VOC2007 when the model is the same. The high $\text{mAP}_{\text{attack}}$ and extremely low $\text{mAP}_{\text{attack+benign}}$ demonstrate that most bboxes changed to the target class have high confidence scores while there are few false positives (bboxes of non-target classes). Furthermore, the average confidence scores of bboxes changing label are >0.86, and $>80\%$ of generated bboxes are all with confidence scores >0.93.

| Attack type | OGA | OGA | RMA | RMA | GMA | GMA | ODA | ODA |
Model	Faster-RCNN	YOLOv3	Faster-RCNN	YOLOv3	Faster-RCNN	YOLOv3	Faster-RCNN	YOLOv3
mAP_{benign} (%)	75.6−	82.1−	72.5−	80.1−	75.6−	81.2−	78.6−	82.2−
AP_{benign} (%)	84.2−	87.9−	83.1−	86.0−	84.3−	86.3−	85.9−	86.8−
mAP_{attack} (%)	74.7⋆	81.6⋆	36.4↑	35.3↑	34.9↑	34.4↑	77.9⋆	81.3⋆
AP_{attack} (%)	87.9↑	90.7↑	36.4↑	35.3↑	34.9↑	34.4↑	-	-
$AP_{attack+benign}$ (%)	-	-	63.1↓	66.2↓	68.6↓	68.3↓	34.4↓	52.1↓
$mAP_{attack+benign}$ (%)	-	-	41.8↓	46.1↓	47.7↓	44.6↓	75.3⋆	80.6⋆
ASR (%)	93.8↑	92.1↑	18.1↑	17.6↑	13.9↑	14.5↑	63.0↑	50.9↑

Table 2: Attack performance after fine-tuning the infected model $F_{infected}$ on another benign dataset $\mathcal{D}'_{train,poisoned}$ and testing for clean and backdoored images from $\mathcal{D}'_{test,poisoned}$. ("↑"/"↓"/"−"/"⋆" follow definitions in Table 1.)

For GMA, the poisoning rate P is 30% and the trigger size $(W_t, H_t) = (49, 49)$. Since there is only one trigger on the left-top corner of the image in GMA, the trigger and target class object(s) may not share the same location, which increases the difficulty of GMA. ASR in GMA is lower than the ASR in RMA when the dataset and model are the same. Besides, the average confidence scores of bboxes changing label are all >0.8, and >80% of generated bboxes are all with confidence score >0.85.

For ODA, the poisoning rate P is 20% and the trigger size $(W_t, H_t) = (29, 29)$. The infected model uses a trigger to offset the object's feature and vanish the target class bbox. The ASR in ODA is lower than ASR in OGA, which shows that learning trigger eliminating object's feature is more complicated than learning trigger's feature. $AP_{attack+benign}$ is low due to disappearance or confidence score decline of target class bboxes. To prove that the infected model uses small triggers to offset objects' features instead of blocking features, we calculate the ASR on the benign model (Faster-RCNN, YOLOv3) with MSCOCO and VOC07+12, and we find all ASR < 5%. In addition, the average confidence scores of vanished bboxes are all <0.22 (if there is no trigger presence, average confidence scores of bboxes with target class are all >0.75), and >80% of vanished bboxes are all with confidence score <0.15.

Transfer Learning Attack. We fine-tune the infected model $F_{infected}$ on a benign training dataset $D'_{train,benign}$ to test whether the hidden backdoor can be removed by transfer learning. To be specific, Faster-RCNN and YOLOv3 are pre-trained on the poisoned MSCOCO, and fine-tuned on the benign VOC2007 (for OGA, RMA, GMA) or benign VOC07+12 (for ODA). In real-world object detection, some people prefer to download a pre-trained model which is trained on a large dataset and fine-tune it on a smaller dataset for specific tasks. It is highly possible that the pre-trained model is trained on a poisoned dataset, and the user fine-tunes it on his own benign, task-oriented dataset. The results of infected model after fine-tuning are in Table 2. All parameters in Table 2 follow the same settings in Table 1. For OGA and ODA, the ASR on "person" target class is high after transfer learning, which implies that fine-tuning on another benign dataset cannot prevent OGA and ODA. For OGA, the model only needs to memorize the pattern of trigger regardless of object's feature. For ODA, the model uses the trigger to offset "person" objects' features.

However, for RMA and GMA, although 80 classes in MSCOCO include 20 classes in VOC2007 (VOC07+12), there exist many classes that the feature of the

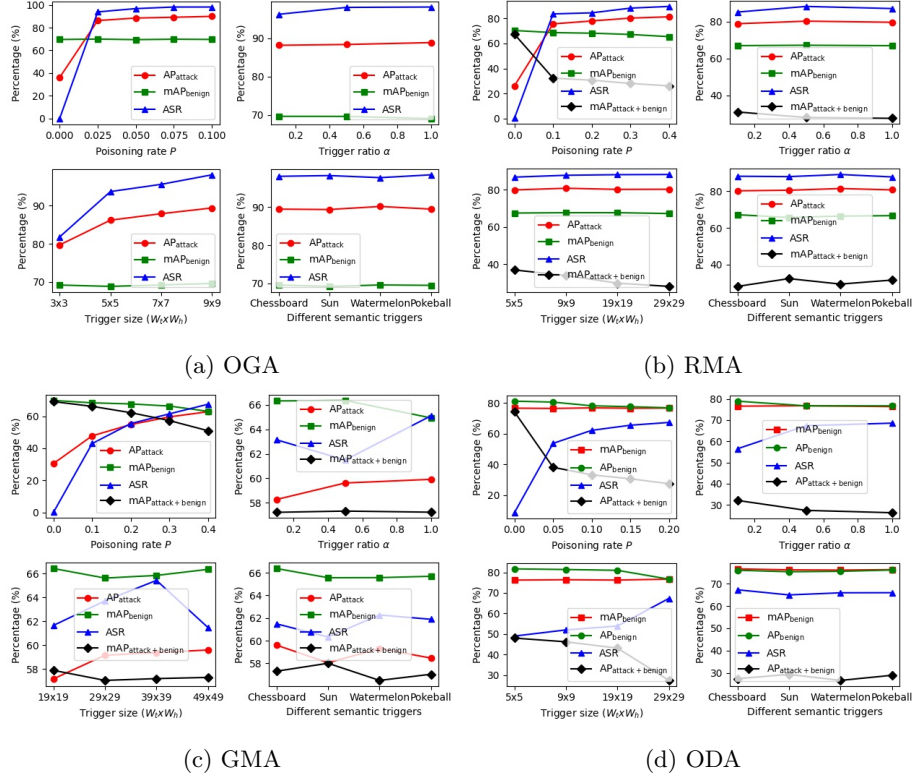(a) OGA   (b) RMA

(c) GMA   (d) ODA

Fig. 3: Impact of parameters and different semantic triggers on various metric for clean and backdoored images.

same class learned from two datasets is different. The trigger alone is not enough to change the class of bbox if the feature learned from two datasets is not similar, which results in poor ASR. For instance, "tv" class in VOC includes various objects like monitor, computer, game, PC, watching, laptop, however, "tv" class in MSCOCO only has television itself. "laptop" and "cellphone" belong to other classes in MSCOCO. Features learned from "tv" class between MSCOCO and VOC are different which explains that only 5% of "tv" objects are changed their classes to target class "person" with confidence score >0.5. While 38% of "car" class objects are changed their class to "person" target class with confidence score >0.5.

5.3 Ablation Study

To explore the different components of our introduced backdoor attacks, we conduct ablation studies on the effects of poisoning rate P, trigger size (W_t, H_t), trigger ratio α, different semantic triggers, target class t, and triggers' locations on backdoor attacks. We use Faster-RCNN on VOC2007 for OGA, RMA, GMA

Attack type	OGA	RMA	GMA	ODA
poisoning rate P (%)	10	30	30	1
$\text{mAP}_{\text{benign}}$ (%)	69.6−	67.5−	63.0−	77.1−
$\text{AP}_{\text{benign}}$ (%)	77.1−	75.2−	71.3−	81.9−
$\text{mAP}_{\text{attack}}$ (%)	70.0⋆	79.9↑	53.0↑	76.9⋆
$\text{AP}_{\text{attack}}$ (%)	98.4↑	79.9↑	53.0↑	-
$\text{AP}_{\text{attack+benign}}$ (%)	-	26.1↓	4.9↓	58.2↓
$\text{mAP}_{\text{attack+benign}}$ (%)	-	25.3↓	52.5↓	76.2∗
ASR (%)	98.7↑	85.2↑	69.4↑	37.2↑

(a) Target class $t =$ "sheep" class.

Attack type	RMA	GMA	ODA
$\text{mAP}_{\text{benign}}$ (%)	67.3−	66.1−	76.8−
$\text{AP}_{\text{benign}}$ (%)	75.1−	74.1−	77.0−
$\text{mAP}_{\text{attack}}$ (%)	80.1↑	57.8↑	76.7⋆
$\text{AP}_{\text{attack}}$ (%)	80.1↑	57.8↑	-
$\text{AP}_{\text{attack+benign}}$ (%)	29.1↓	58.5↓	27.3↓
$\text{mAP}_{\text{attack+benign}}$ (%)	29.5↓	58.1↓	74.3∗
ASR (%	88.3↑	58.9↑	67.8↑

(b) Random triggers' locations.

Table 3: Attack performance when (a) target class t changed to "sheep" class and (b) trigger's locations changed to random locations. ("↑"/"↓"/"−"/"⋆" follow definitions in Table 1.)

and VOC07+12 for ODA. All parameters used in this section are same as parameters used in Table 1. Only one parameter is modified in each ablation study to observe its effects.

From Fig. 3, we find that 1) the poisoning rate P controls the number of poisoned training images, which heavily influences the ASR and other metrics for all settings; 2) a larger trigger size (W_t, H_t) contributes to better attack performance of OGA, ODA; 3) a higher trigger ratio α marginally impacts ASR and other metrics of OGA, RMA, GMA. For OGA, RMA, GMA, the adversary could use a minimal trigger ratio $\alpha = 0.1$ to make the trigger almost invisible on the image. For RMA, the adversary can use an extremely small trigger (5×5) to get a decent attack performance and make the trigger hard to detect. Furthermore, metrics from different semantic triggers are almost the same, which demonstrates the generalizability of using various triggers.

We also change the target class t from "person" (class with most objects) to "sheep" (class with fewer objects). See Appendix C for more detailed statistics. In Table 3 (a), fewer target class objects do not affect the performance of OGA, RMA, GMA. However, ODA obtains poor results since it requires more target class objects to get good attack result. The ASR of ODA on benign model is 4.7%, which proves the infected model learns to vanish "sheep" object instead of blocking object feature by trigger.

To prove that the trigger's location does not influence attack results, we change the trigger's location to a random location in the poisoned dataset and the attacked dataset. For RMA and ODA, trigger's location changes to a random location inside the bbox rather than the left-top corner of the bbox. For GMA, trigger's location is a random location on the image rather than the left-top corner of the image. Table 3 (b) shows results with random location, which are similar to those in Table 1.

6   Detector Cleanse

We propose a detection method: **Detector Cleanse** to identify poisoned testing samples from four attack settings for any deployed object detector. Most defense/detection methods from the backdoor attacks on image classification

cannot apply to object detection. Methods that predict the distributions of backdoor triggers through generative modeling or neuron reverse engineering [37,25,39,15,31] assume the model is a simple neural network instead of multiple parts. Besides, the output of the object detection model (numerous objects) is different from the image classification model (predicted class). Pruning methods [19] remove neurons with low activation rate on the benign dataset and observe the change of $\text{mAP}_{\text{benign}}$ and ASR. However, the pruning method requires high training costs and assumes the user has access to the attacked dataset and understands the adversary's goal. Moreover, pruning some object detection models lead to a moderate drop in performance (mAP) [7,36].

Only some methods such as STRIP [6] and one-pixel signature [13] can generalize to this task but lead to poor performance. For example, in the Faster-RCNN + VOC2007 setting, we modify STRIP to calculate the average entropy of all predicted bboxes. When we set the False Rejection Rate (FRR) to 5%, the False Acceptance Rate (FAR) is $\geq 30\%$ on four attack settings. The vanilla classifier from one-pixel signature only successfully classifies 17 models among 15 clean and 15 backdoor models. Moreover, these methods have strong assumptions: STRIP assumes the user has access to a subset of clean images, and one-pixel signature supposes the user has a clean model or clean dataset, making them less practical to defend against BadDet.

Since previous methods cannot be generalized to object detection, we propose **Detector Cleanse**, a run-time poisoned image detection framework for object detectors, which assumes the user only has a few clean features (can be drawn from different datasets). The key idea is that the feature of the small trigger has a single (strong) input-agnostic pattern. Even though strong perturbation is applied on a small region in the predicted bbox, the poisoned detector still behaves as the attacker specifies on the target class. And this behavior is abnormal, making it possible to detect backdoor attacks. Given a perturbed region with features from different classes, the probability of various classes on the predicted bbox should vary. In particular, the target class's predicted bboxes on OGA, RMA, GMA should have small entropy. And target class's predicted bbox on ODA should generate larger entropy because the trigger offsets the correct class's feature and decreases the highest predicted class's probability. A more balanced class's probability distribution should generate larger entropy.

For four attack settings, we have tested 500 clean images and 500 poisoned images from VOC2007 testing set on Faster-RCNN. The poisoned model is trained by the same setting in Table 1. The detailed algorithm is shown in Appendix E. Define two hyperparameters: detection mean m and detection threshold Δ. Given each image x, $N = 100$ features $\chi = \{x_1, \ldots, x_N\}$ are drawn from a small portion of clean VOC2007 ground-truth bboxes (We can also use clean features from different datasets. Appendix F shows features from MSCOCO get similar results). Then, for each predicted bbox b on x, the feature is linearly blended with chosen bbox region on x to generate $N = 100$ perturbed bboxes, and we calculate the average entropy of these bboxes. If the average entropy

Attack Type	$\Delta = 0.25$			$\Delta = 0.3$			$\Delta = 0.35$		
	Accuracy	FAR	FRR	Accuracy	FAR	FRR	Accuracy	FAR	FRR
OGA	87.5%	**2.7%**	9.8%	91.0%	4.1%	4.9%	**91.3%**	6.3%	**2.4%**
RMA	85.0%	**4.9%**	10.1%	88.6%	6.2%	5.2%	**90.2%**	7.5%	**2.3%**
GMA	80.4%	**9.6%**	10.0%	82.6%	12.3%	5.1%	**83.3%**	14.2%	**2.5%**
ODA	83.5%	**6.3%**	10.2%	87.3%	7.7%	5.0%	**88.6%**	9.0%	**2.4%**

Table 4: Results of **Detector Cleanse** on Faster-RCNN + VOC2007 (Detection mean $m = 0.51$, The best scores in same Attack Type are set in **bold**)

doesn't fall in the interval $[m - \Delta, m + \Delta]$, we mark the corresponding image as poisoned and return the bbox's coordinate to identify the trigger's position.

To evaluate the performance of **Detector Cleanse**, we calculate Accuracy, FAR and FRR on four attack types in Table 4. Since we assume the user has no access to poisoned samples and only has a few features (N) from the benign bboxes' regions, the user can only use those features to estimate the entropy distribution of benign bboxes. The user assumes the distribution is normal, and then the user calculates the mean (0.55) and standard deviation (0.15) of entropy distribution from features. Finally, we set m to mean of entropy distribution and Δ around double standard deviation on all settings. For metric FRR and FAR, FAR is the probability that all bboxes' entropy on poisoned image falls in the interval $[m - \Delta, m + \Delta]$; FRR is the probability of at least one bbox's entropy on the clean image is smaller than $m - \Delta$ or larger than $m + \Delta$. Theoretically, we can control FRR by setting Δ corresponding to standard deviation. From Table 4, Δ determines FRR, and FRR becomes smaller and FAR becomes larger as Δ increases. If the security concern is serious, the user can set a smaller detection threshold Δ to get a smaller FAR and larger FRR. The FAR from RMA, GMA is high because sometimes the detector generates target class bbox with a low confidence score. For ODA, failing to decrease the confidence score of the target class bbox causes high FAR.

7   Conclusion

This paper introduces four backdoor attack methods on object detection and defines appropriate metrics to evaluate the attack performance. The experiments show the success of four attacks on two-stage (Faster-RCNN) and one-stage (YOLOv3) models and demonstrate that transfer learning cannot entirely remove the hidden backdoor in the object detection model. Furthermore, the ablation study shows the influence of each parameter and trigger. We also propose **Detector Cleanse** framework to detect whether an image is poisoned given any deployed object detector. In conclusion, object detection is commonly used in real-time applications like autonomous driving and surveillance, so the infected object detection model, which often integrates into an extensive system, will pose a significant threat to real-world applications.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate (2016)
2. Chen, X., Liu, C., Li, B., Lu, K., Song, D.: Targeted backdoor attacks on deep learning systems using data poisoning (2017)
3. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9185–9193 (2018)
4. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results (2007)
5. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results (2012)
6. Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D.C., Nepal, S.: Strip: A defence against trojan attacks on deep neural networks. In: Proceedings of the 35th Annual Computer Security Applications Conference. pp. 113–125 (2019)
7. Ghosh, S., Srinivasa, S.K.K., Amon, P., Hutter, A., Kaup, A.: Deep network pruning for object detection. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 3915–3919 (2019). https://doi.org/10.1109/ICIP.2019.8803505
8. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (2015)
9. Graves, A., Mohamed, A., Hinton, G.E.: Speech recognition with deep recurrent neural networks. CoRR **abs/1303.5778** (2013), `http://arxiv.org/abs/1303.5778`
10. Gu, T., Dolan-Gavitt, B., Garg, S.: Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733 (2017)
11. Gu, T., Liu, K., Dolan-Gavitt, B., Garg, S.: Badnets: Evaluating backdooring attacks on deep neural networks. IEEE Access **7**, 47230–47244 (2019). https://doi.org/10.1109/ACCESS.2019.2909068
12. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 346–361. Springer International Publishing, Cham (2014)
13. Huang, S., Peng, W., Jia, Z., Tu, Z.: One-pixel signature: Characterizing cnn models for backdoor detection. In: European Conference on Computer Vision. pp. 326–341. Springer (2020)
14. Kurita, K., Michel, P., Neubig, G.: Weight poisoning attacks on pretrained models. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 2793–2806. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.249, `https://aclanthology.org/2020.acl-main.249`
15. Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., Ma, X.: Neural attention distillation: Erasing backdoor triggers from deep neural networks. In: International Conference on Learning Representations (2021), `https://openreview.net/forum?id=9l0K4OM-oXE`
16. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
17. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)

18. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., Zitnick, L.: Microsoft coco: Common objects in context. In: ECCV. European Conference on Computer Vision (September 2014), `https://www.microsoft.com/en-us/research/publication/microsoft-coco-common-objects-in-context/`
19. Liu, K., Dolan-Gavitt, B., Garg, S.: Fine-pruning: Defending against backdooring attacks on deep neural networks. In: RAID (2018)
20. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
21. Liu, Y., Ma, X., Bailey, J., Lu, F.: Reflection backdoor: A natural backdoor attack on deep neural networks. In: European Conference on Computer Vision. pp. 182–199. Springer (2020)
22. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning (2013)
23. Nguyen, T.A., Tran, A.: Input-aware dynamic backdoor attack. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 3454–3464. Curran Associates, Inc. (2020), `https://proceedings.neurips.cc/paper/2020/file/234e691320c0ad5b45ee3c96d0d7b8f8-Paper.pdf`
24. Nguyen, T.A., Tran, A.T.: Wanet - imperceptible warping-based backdoor attack. In: International Conference on Learning Representations (2021), `https://openreview.net/forum?id=eEn8KTtJOx`
25. Qiao, X., Yang, Y., Li, H.: Defending neural backdoors via generative distribution modeling. In: NeurIPS (2019)
26. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
27. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement (2018). arXiv preprint arXiv:1804.02767 **20** (2018)
28. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 28. Curran Associates, Inc. (2015), `https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf`
29. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge (2015)
30. Saha, A., Subramanya, A., Pirsiavash, H.: Hidden trigger backdoor attacks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 11957–11965 (2020)
31. Shen, G., Liu, Y., Tao, G., An, S., Xu, Q., Cheng, S., Ma, S., Zhang, X.: Backdoor scanning for deep neural networks through k-arm optimization. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 9525–9536. PMLR (2021), `http://proceedings.mlr.press/v139/shen21c.html`
32. Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D.: Mastering the game of go

with deep neural networks and tree search. Nature **529**, 484–489 (01 2016). https://doi.org/10.1038/nature16961

33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
34. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: International Conference on Learning Representations (2014)
35. Turner, A., Tsipras, D., Madry, A.: Label-consistent backdoor attacks (2019)
36. Tzelepis, G., Asif, A., Baci, S., Cavdar, S., Aksoy, E.E.: Deep neural network compression for image classification and object detection (2019)
37. Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., Zhao, B.Y.: Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In: 2019 IEEE Symposium on Security and Privacy (SP). pp. 707–723 (2019). https://doi.org/10.1109/SP.2019.00031
38. Xie, C., Huang, K., Chen, P.Y., Li, B.: Dba: Distributed backdoor attacks against federated learning. In: ICLR (2020)
39. Xu, K., Liu, S., Chen, P.Y., Zhao, P., Lin, X.: Defending against backdoor attack on deep neural networks (2021)
40. Zhao, S., Ma, X., Zheng, X., Bailey, J., Chen, J., Jiang, Y.G.: Clean-label backdoor attacks on video recognition models. pp. 14431–14440 (06 2020). https://doi.org/10.1109/CVPR42600.2020.01445
41. Zou, Z., Shi, Z., Guo, Y., Ye, J.: Object detection in 20 years: A survey. ArXiv **abs/1905.05055** (2019)