# Exploring the Vulnerability of Single Shot Module in Object Detectors via Imperceptible Background Patches

Yuezun Li[1]
yli52@albany.edu
Xiao Bian[2]
xiao.bian@ge.com
Ming-Ching Chang[1]
mchang2@albany.edu
Siwei Lyu[1]
slyu@albany.edu

[1] University at Albany,
State University of New York, USA
[2] GE Global Research Center,
Niskayuna, New York, USA

**Abstract**

Recent works succeeded to generate adversarial perturbations on the entire image or the object of interests to corrupt CNN based object detectors. In this paper, we focus on exploring the vulnerability of the Single Shot Module (SSM) commonly used in recent object detectors, by adding small perturbations to patches in the background outside the object. The SSM is referred to the Region Proposal Network used in a two-stage object detector or the single-stage object detector itself. The SSM is typically a fully convolutional neural network which generates output in a single forward pass. Due to the excessive convolutions used in SSM, the actual receptive field is larger than the object itself. As such, we propose a novel method to corrupt object detectors by generating imperceptible patches only in the background. Our method can find a few background patches for perturbation, which can effectively decrease true positives and dramatically increase false positives. Efficacy is demonstrated on 5 two-stage object detectors and 8 single-stage object detectors on the MS COCO 2014 dataset. Results indicate that perturbations with small distortions outside the bounding box of object region can still severely damage the detection performance.

## 1 Introduction

Convolutional Neural Networks (CNN) are shown to be vulnerable against *adversarial perturbations* [10], which are intentionally designed and imperceptible noise added to the input that can drastically affect network performance. Many works [2, 6, 10, 14, 23, 24, 25, 26, 31, 34] have investigated this vulnerability and proposed various adversarial attack methods to impair image classifiers.

Recently, adversarial perturbations are extended to networks for other computer vision tasks such as the object detectors and semantic/instance segmentation networks [4, 7, 15, 21, 22, 33]. However, all existing methods focus on creating adversarial perturbations on either the entire image or the object itself. An intuitive question to ask is: *can adversarial perturbations be added solely on the background to achieve similar vulnerability?* We will address this very problem in this paper. Specifically, we explore the vulnerability of the single shot feedforward network in the state-of-the-art two-stage [29] and single-stage object detectors [16, 18, 19, 28] and show that the mechanism can be corrupted by adding imperceptible perturbations on a few small background patches, which can not only decrease true positives of detected objects, but also increase false positives in the background. Figure 1 shows a visual illustration of this approach. Looking forward, how to address the vulnerability of modern object detectors explored in this work will be a critical open issue, as applications including autonomous driving and AI medical image analysis demand highly reliable and trustworthy object detectors [1].

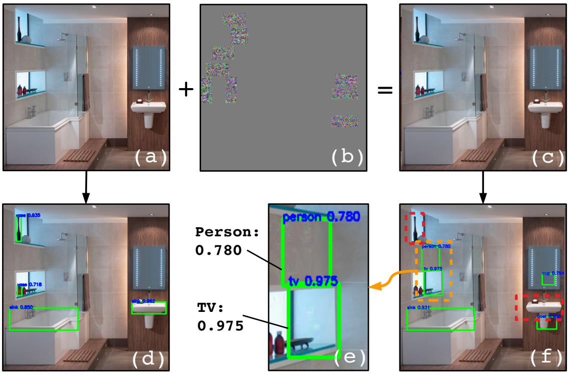

Figure 1: *Visual illustration of the background patch attack on object detectors. (a) Original image. (b) Adversarial background patches, amplified by a factor of 30 for visualization. (c) Perturbed image. (d, f) Detection results of (a, c) using Faster-RCNN [29] respectively. (e) Zoom-in of the false positives "person" and "TV" in (d). Red boxes in (d) denote miss detections.*

Mainstream CNN based object detectors call into two categories: *two-stage* and *single-stage* object detectors. The **Single Shot Module (SSM)** refers to the Region Proposal Network (RPN) in the two-stage object detectors or the single-stage object detector itself. Since the SSM makes use of excessive convolutions in multiple layers, the receptive field is often much larger than the object size. This is precisely how contextual information outside the objects can be leveraged to improve detection. However, this property also makes SSM vulnerable for attacks coming from the background. If SSM was corrupted, no correct object proposals or detections will be generated, therefore leading to large errors.

In this paper, we propose a novel method to generate adversarial background patches to attack SSM. Our method finds effective locations and shape for background patches to create adversarial perturbations inside, which not only decreases the true positives, but also dramatically increases the false positives in the background. To the best of our knowledge, most existing works focused on disrupting the true positives, and very few address the false positives — which, in our view, is equally important regarding vulnerability in practical use, see Figure 1(e). Our method can effectively corrupt the SSM output ranking, such that false positives can be pushed ahead of true positives, see Figure 2(d-e). Our adversarial background patch attack aims at achieving the following three aspects: (1) decreasing the classification scores of correct detections, (2) corrupting the shape offset regression which shifts the localization (shape and location) of the correct detections, and (3) increasing the (non-background) object class scores that should not come up in the background. Our background patch attack generation can be cast as an optimization problem by minimizing the combination of the following three loss terms: (1) *True Positive Class (***TPC***) loss*, which characterizes the correct class scores of the true positives; (2) *True Positive Shape (***TPS***) loss*, which represents the correctness of shape offset regression of the true positives; (3) *False Positive Class (***FPC***) loss*, which characterizes the non-background class scores of

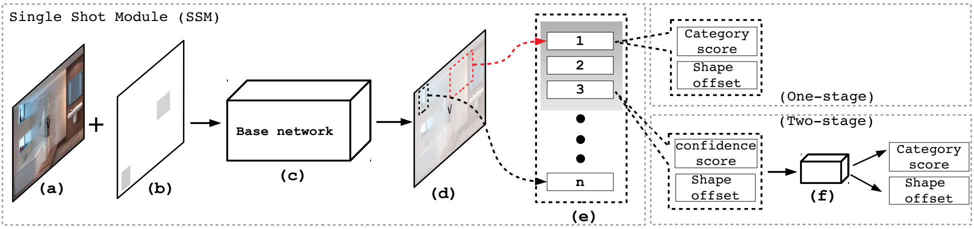

Figure 2: *Overview. (a) Original image. (b) Background patches generated by our method. (c) Base-network, which is the RPN for two-stage object detectors or the single-stage object detector itself. (d) Output of SSM, where the red box and black box denote a false positive and a true positive, respectively. Our attack can disrupt the top ranked results by decreasing true positives and increasing false positives. (e) denotes the top ranked results, which are the object proposals for two-stage object detectors or the detections for single-stage object detectors. (f) Sub-network of two-stage object detectors for class labels prediction and shape refinements.*

false positive arisen from the background. The combination of these three loss terms can be minimized using iterative gradient descent, such that the desired quantities regarding adversarial background patches (pixels, location and shape) can be calculated.

To demonstrate the efficacy of our method, we conduct experiments on the MS COCO 2014 dataset [17] by attacking 5 mainstream two-stage object detectors and 8 single-stage object detectors. We conducted ablation studies investigating the vulnerability effects with respect to the following factors, and show how they can affect the degraded performance: (1) distance between generated background patches and the object of interest, (2) scale of the object to be detected, and (3) distances between objects.

The contributions can be summarized as four-fold:

- To the best of our knowledge, we are the first to explore the vulnerability of two-stage and single-stage object detectors by adding imperceptible adversarial perturbations on small patches in background.
- Our background patch attack can effectively decrease true positives and increase false positives in the background.
- We conduct comprehensive experiments on mainstream object detectors (5 two-stage and and 8 single-stage ones) to expose their vulnerability.
- Our method can generate 'targeted' false positives of a given class in the background (§4.2), which can cause serious vulnerability, *e.g.*, to force a autonomous driving detector to trigger false pedestrian detections.

## 2 Related Works

**Object Detectors.** The objects of interest in an image are detected by producing bounding boxes and class labels. The state-of-the-art object detectors are deep neural network based, where the network architecture consists of either a single-stage forward pass [16, 18, 19, 28] or a two-stage pipeline [8, 9, 29]. The two-stage object detector first generates object proposals, and then predict class labels and refine the shapes and locations of the proposals. Faster-RCNN [29] is the very recent two-stage object detector, which improves the detection efficiency by using a Region Proposal Network (RPN) to generate object proposals. The RPN is essentially a fully convolutional neural network (FCN), which generates all object proposals in a single forward pass. First a set of anchor boxes are identified, then object proposals are generated by estimating the location and shape of each anchor box with confidence scores. The top ranked object proposals will be selected for further classification and refinement. RPNs are effective and widely used in current two-stage object detectors. In comparison, single-stage object detectors [16, 18, 19, 28] can be viewed as an upgraded version of RPNs, where all detections are generated in a single forward pass (without generating proposals). Instead of predicting the confidence scores of the object proposals in RPN, single-stage object detectors directly predict the classification score for each detection.

**Adversarial Attacks for Image Classifiers.** Adversarial perturbations are intentionally designed noises that are imperceptible to human observers, yet can seriously reduce the deep neural network performance if added to the input image. Many methods [2, 10, 14, 23, 24, 25, 26, 31, 34] have been proposed to impair image classifiers by adding adversarial perturbations on the entire image. Recently, [3, 6] show that adversarial attacks can be generated in the physical world, using printable "stickers" that can be put on the objects in the scene to fool image classifiers. These stickers are clearly identifiable from human eyes but not the machines. The work of [13] generates an artificial adversarial patch in the background that is notable to human eyes but can cripple machine image classification. In contract, our method provides more sophisticated attacks on object detection, and the perturbation is imperceptible to human eyes.

**Adversarial Attacks for Object Detectors.** Recent research also explores the vulnerability of object detectors with extended network architectures [4, 7, 15, 21, 33]. Object detection is widely used in practical applications such as autonomous driving, thus the impact of attack *vs.* vulnerability is greater. In [21], adversarial perturbations are added to the stop-sign and face images that can cripple their detections. The dense adversary generation in [33] iteratively impairs both object detection and semantic segmentation. A physical adversarial perturbation method is proposed in [4] to attack Faster-RCNN based stop-sign detector. The work [7] extends [4] to other detectors such as YOLO, in that a physical adversarial sticker can cripple the stop-sign detection. The robust adversarial perturbations in [15] corrupts deep proposal-based object detectors and instance segmentation methods by attacking the RPN, thus only two-stage networks are attacked. All aforementioned methods focus on adding adversarial perturbations on the entire image or the object itself. Moreover, these methods have no intention to attack on the aspect of increasing false positives, their goal is solely on decreasing the true positives. The work [20] generated a visible adversarial patch on the left-top corner of image to fool Faster-RCNN and YOLO. In contrast, our adversarial attack is more universal in that: (1) most modern object detectors including both single-stage and two-stage ones are covered, (2) both the true and false positives of the detection are explicitly impaired, and (3) the added perturbation is imperceptible and only on the small patches that are optimally selected in background.

# 3 Methods

Our method can generate imperceptible background patches, which can effectively damage mainstream CNN object detectors, by simultaneously decreasing true positives and increasing false positives in background. The true positives will be corrupted with decreased classification score and largely shifted localization (shape and location). The false positives will be exacerbated with increased (non-background) class scores that should not come up in the background. Our approach can be formulated as the minimization of three loss terms: *True Positive Class (***TPC***) loss*, *True Positive Shape (***TPS***) loss* and *False Positive Class (***FPC***) loss*, which we will formally define in §3.2-3.4. The loss minimization can be computed using iterative gradient descent in §3.5, where the variables of unknown are the pixel perturbations and localization (shape and location) of the background patches.

Our attack method works in a 'white-box' manner, *i.e.*, we assume that it has access to network parameter for back-propagation gradient computation.

## 3.1 Notations and Problem Formulation

Let $\mathcal{I}$ denote the input image, $\{\bar{b}_i = (\bar{x}_i, \bar{y}_i, \bar{w}_i, \bar{h}_i), i = 1, \ldots, N\}$ denote the $N$ ground truth bounding boxes $\{\bar{b}_i\}$ for the objects of interest in image $\mathcal{I}$, where $(\bar{x}_i, \bar{y}_i)$ are the box center coordinates, $(\bar{w}_i, \bar{h}_i)$ are their widths and heights, respectively. Let $\{0, \ldots, C\}$ denote the set of class labels, where 0 is the background class, and $C+1$ is the number of classes. Specifically, for two-stage object detectors, $C = 1$ as the focus is to attack RPN, only the two classes of background/object need to be considered. [1]

Let $\mathcal{F}$ denote the Single Shot Module (SSM) with fixed model parameters. Let $\mathcal{F}(\mathcal{I}) = \{(s_j^c, b_j), j = 1, \ldots, M\}$ denote the SSM results of either RPN object proposals or single-stage detections on image $\mathcal{I}$ [2], where $s_j^c$ denote the score of class $c$ after softmax, $b_j$ denote the bounding box of the $j$-th detection, and $M$ is the number of SSM proposals/detections. Let $b_j = (x_j, y_j, w_j, h_j)$, where $(x_j, y_j)$ are the box center coordinates, and $(w_j, h_j)$ are their widths and heights, respectively. Let $\mathcal{Q} = \{(\tilde{x}_k, \tilde{y}_k, \tilde{w}_k, \tilde{h}_k), k = 1, \ldots, K\}$ denote the adversarial background patches, with locations specified by $(\tilde{x}_k, \tilde{y}_k)$ and shapes specified by $(\tilde{w}_k, \tilde{h}_k)$; $K$ is the number of background patches that will be added for attack. Let $\mathcal{I} \odot \mathcal{Q}$ denote the masked pixel regions on image $\mathcal{I}$ specified by the background patches $\mathcal{Q}$.

Our goal is to generate background patches with small pixel changes that can disrupt SSM. The adversarial attack is then the search of both the background patch geometry (location, size and shape) and pixel changes to be altered. We formulate this optimization as the minimization of three loss terms: TPC denoted as $L_{tpc}$, TPS as $L_{shape}$, and FPC as $L_{fpc}$. To maximize adversarial attack effects, while minimizing any distortion to the input image $\mathcal{I}$, we control the amount of pixel change inside the background patches $\mathcal{Q}$, by employing the *Peak Signal-to-Noise Ratio* (PSNR), a widely used metric for human perception of image quality. Smaller distortion leads to higher PSNR value. Specifically, the adversarial background patch can be produced by minimizing the following loss function *w.r.t.* $\mathcal{I} \odot \mathcal{Q}$, considering the location and shape of the background patches $\mathcal{Q}$ and the included pixel value $\mathcal{I} \odot \mathcal{Q}$ as variables:

$$\begin{aligned} &\min_{\mathcal{I} \odot \mathcal{Q}} \; \left\{ L_{tpc}(\mathcal{I} \odot \mathcal{Q}; \mathcal{F}) + L_{shape}(\mathcal{I} \odot \mathcal{Q}; \mathcal{F}) + L_{fpc}(\mathcal{I} \odot \mathcal{Q}; \mathcal{F}) \right\}, \\ &\text{s.t. PSNR}(\mathcal{I} \odot \mathcal{Q}) \geq \varepsilon, \end{aligned} \tag{1}$$

where $\varepsilon$ is the lower bound of PSNR. Compared to a recent work [15] which creates adversarial perturbations on the entire image to decrease true positives, our method creates adversarial perturbations only in selected background patches and can largely increase false positives as well.

## 3.2 True Positive Class (TPC) Loss

Our approach attacks detectors by only introducing changes in the background. Since the sum of all class scores is 1, the attack of decreasing the score of the correct class $c$ can

[1] Note that our method does not require the original ground truth labeling (used for detector training). In practice, we can use the first test detection results as the ground truth to compute the three loss terms.

[2] We denote object proposals as detections hereafter for simplicity.

be alternatively achieved by increasing the score of another running up class $\hat{c}$, such that $s^{\hat{c}} > s^{c}$ invalidates a good detection. To make this attack most effective, we try to increase the score of the running up class $\hat{c}$ with the largest score $s^{\hat{c}}$ among all classes except $c$, *i.e.*, $\hat{c} \in \{0,...,C\}/\{c\}$. This running up detection selection considers the following criteria: (1) detection $b_j$ with IoU overlapping with its ground truth box greater than threshold 0.5, and (2) class score $s^{c}_{j}$ greater than threshold 0.1. Let $z_j = 1$ for the detection $b_j$ satisfying the above two criteria, and $z_j = 0$ otherwise. The TPC loss $L_{tpc}$ sums up the cross entropy of scores from the selected running up detections $z_j$ among all $M$ detections:

$$L_{tpc}(\mathcal{I} \odot \mathcal{Q}; \mathcal{F}) = -\sum_{j=1}^{M} z_j \log(s^{\hat{c}}_{j}). \tag{2}$$

Minimizing Eq.(2) increases score $s^{\hat{c}}$ from an incorrect class that can effectively invalidates true positives.

## 3.3 True Positive Shape (TPS) Loss

Shape regression is an important step to refine the localization of detections (or proposals), where the locations and shapes of anchor boxes are adjusted to match the corresponding ground truth boxes, by expressing the localization in terms of *offsets*. In general, CNN object detectors are vulnerable under attack at the shape regression step, even when the classification is functioning perfectly, as good detections will be pushed away from their desired locations. The TPS loss is designed to push away the predicted localization from the correct ones. Let $\Delta x_j, \Delta y_j, \Delta w_j, \Delta h_j$ denote the predicted offset in terms of object center and bounding box size. Let $\Delta\bar{x}_j, \Delta\bar{y}_j, \Delta\bar{w}_j, \Delta\bar{h}_j$ denote the true offset between the corresponding anchor boxes and ground truth boxes. The TPS loss $L_{shape}$ sums up the squared offset differences of selected true positives under the criteria $z_j$ defined in § 3.2 among all $M$ detections:

$$\begin{aligned} L_{shape}(\mathcal{I} \odot \mathcal{Q}; \mathcal{F}) = \exp\{-\textstyle\sum_{j=1}^{M} z_j \cdot \\ \big[(\Delta x_j - \Delta\bar{x}_j)^2 + (\Delta y_j - \Delta\bar{y}_j)^2 \\ +(\Delta w_j - \Delta\bar{w}_j)^2 + (\Delta h_j - \Delta\bar{h}_j)^2\big]\}. \end{aligned} \tag{3}$$

Minimizing Eq.(3) encourages pushing the predicted offsets $\Delta x_j, \Delta y_j, \Delta w_j, \Delta h_j$ away from the true offsets $\Delta\bar{x}_j, \Delta\bar{y}_j, \Delta\bar{w}_j, \Delta\bar{h}_j$, to corrupt the predicted localization of $b_j$. Note that in contrast to [15] which disrupts the shape offset regression by pushing the localization toward a large constant value, here we directly optimize against known ground truth values, which should be more effective.

## 3.4 False Positive Class (FPC) Loss

We introduce FPC as a novel loss term to strengthen the attack that can corrupt detectors by increasing false positives in the background. In the case without attack, the background should only contain detections (or proposals) with high scores belonging to the background class. To make detectors generate false positives, the attack should make the score of an object class $c' \in \{1,...,C\}$ greater than that of the background class 0, (*i.e.*, $s^{c'} > s^{0}$), to push the incorrect detections ahead to the top. The red box in Figure 2 (d-e) shows one such example. To make such attack most effective, we propose to try pushing forward the class instance $c'$ with the largest score among $\{1,...,C\}$ in the FPC loss design. Specifically, since the goal is to create false positives in the background, only detections $b_j$ satisfying the following conditions need to be considered: (1) detection $b_j$ without overlapping with any ground truth box (*i.e.*, fully in the background), and (2) detection $b_j$ with IoU overlapping with the generated background patches $\mathcal{Q}$ greater than threshold 0.1. Let $r_j = 1$ for the detection $b_j$ satisfying the above criteria, and $r_j = 0$ otherwise. The FPC loss $L_{fpc}$ sums up the cross entropy of the selected $r_j$ scores among all $M$ detections:

$$L_{fpc}(\mathcal{I} \odot \mathcal{Q}; \mathcal{F}) = -\sum_{j=1}^{M} r_j \log(s_j^{c'}). \tag{4}$$

Minimizing Eq.(4) encourages to increase the score $s_j^{c'}$ of some incorrect object class in the background, which thereby creates false positives.

## 3.5 Background Patches Generation

Our method generates and refines the adversarial background patches in the overall loss optimization iterations. We describes how the background patches are initialized, and how they can be expanded and refined, according to the framework in Eq.(1) incorporating the three loss terms. Direct minimization of the loss function in Eq.(1) with respect to $\mathcal{I} \odot \mathcal{Q}$ is difficult. Thus, we employ a standard iterative optimization scheme using gradient descent.

In general, an object detector finds multiple objects, and the closer the objects are, the greater their receptive fields overlap. This suggests that a single adversarial background patch can corrupt the detection of multiple objects, as long as the background patch are close enough to them. To select where best to put in background patches for most effective adversarial attack, we consider the spatial distribution of the objects and the potential locations, shapes, and sizes of background patches. We start with clustering the objects of interest into groups based on their spatial distances within the image. For each group, we empirically generate $n_b = 3$ background patches as initialization.

Algorithm 1 lists the pseudo code of our adversarial image generation procedure. We first compute the gradient of the overall loss term *w.r.t.* $\mathcal{I}_t$ as

$$\mathcal{G}_t = \nabla_{\mathcal{I}_t} \left[ L_{tpc}(\mathcal{I}_t; \mathcal{F}) + L_{shape}(\mathcal{I}_t; \mathcal{F}) + L_{fpc}(\mathcal{I}_t; \mathcal{F}) \right] \tag{5}$$

where $t$ denote the iteration number. [3] We next describe how the background patches $\mathcal{Q}$ are initialized and updated.

**Initial background patches:** We consider candidates of background patches for each targeted object, with size initialized to 0.2 of the object size, and aspect ratios (1, 0.67, 0.75, 1.5, 1.33). Sliding windows are used to select, for each object group, the best location and shape of the background patches $\mathcal{Q}_0$, according the the following criteria: (1) The distance between background patch and objects should not be less than a threshold, as 0.2 of the largest object box side. (2) The patch with largest sum of $\mathcal{G}_t$ gradient intensities is preferred. (3) No selected patches should overlap. Such patch selection repeats until $n_b$ background patches are obtained for each group.

**Expanding background patches:** In the subsequent iterations $t > 0$, $\mathcal{Q}_t$ expends from $\mathcal{Q}_{t-1}$ with a small stride (0.02 of the shorter side of $\mathcal{I}$) in one of the 4 possible directions (left, right, top, down). The extension direction is determined by the one where $\mathcal{G}_t$ gradient intensity in $\mathcal{Q}_t$ increases the most. In our method the background patch only expands (no shrinking) in the iterations until termination.

[3] Note that we omit the $\mathcal{Q}$ term from Eq.(1) in this gradient formula, as $\mathcal{I}_t$ implies the $\mathcal{I} \odot \mathcal{Q}$ masking has already been performed in the iterations.

**Algorithm 1** *Background Patch Generation*

```
Require: SSM model $\mathcal{F}$; input image $\mathcal{I}$; maximal iteration $T$
$\mathcal{I}_0 = \mathcal{I}, t = 0$
while $t < T$ and $\sum_{j=1}^{M} z_j \neq 0$ do
    $\mathcal{G}_t = \nabla_{\mathcal{I}_t} \left[ L_{tpc}(\mathcal{I}_t;\mathcal{F}) + L_{shape}(\mathcal{I}_t;\mathcal{F}) + L_{fpc}(\mathcal{I}_t;\mathcal{F}) \right]$
    if $t = 0$ then
        $\mathcal{Q}_0 \leftarrow$ initial background patches
    else
        $\mathcal{Q}_t \leftarrow$ expanded background patches
    $\mathcal{P}_t = \mathcal{G}_t \odot \mathcal{Q}_t$
    $\hat{\mathcal{P}}_t = \frac{\lambda}{||\mathcal{P}_t||_2} \cdot \mathcal{P}_t$
    $\mathcal{I}_{t+1} = \text{clip}(\mathcal{I}_t - \hat{\mathcal{P}}_t)$
    if PSNR$(\mathcal{I}_{t+1} \odot \mathcal{Q}_t) < \varepsilon$ then
        break
    $t = t + 1$
Ensure: Adversarial perturbed image $\mathcal{I}_t$
```

The adversarial image perturbations at iteration $t$ is denoted as $\mathcal{P}_t$, which can be calculated as the intersection of the gradient image $\mathcal{G}_t$ of the overall loss and the current background patches $\mathcal{Q}_t$, *i.e.*, $\mathcal{P}_t = \mathcal{G}_t \odot \mathcal{Q}$. A L2 normalized perturbation $\hat{\mathcal{P}}_t$ is then calculated to update the adversarial image $\mathcal{I}_t$, using scale parameter $\lambda = 30$. The adversarial perturbed image $\mathcal{I}_t$ is then clipped into $[0, 255]$.

The optimization iteration continues until any of the following condition is reached: (1) maximal iteration $T = 250$ is reached, (2) no true positive selection are available for TPS and TPS, *i.e.*, $\sum_{j=1}^{M} z_j = 0$ or (3) the RSNR$(\mathcal{I} \odot \mathcal{Q})$ is less than a maximal image distortion threshold $\varepsilon$. Since the PSNR in lossy image compression is typically between 30 to 50 dB [32], we empirically set $\varepsilon = 35$ dB for two-stage object detectors and $\varepsilon = 30$ dB for single-stage ones.

# 4 Experiments

We perform experimental evaluations of the proposed adversarial attacks on mainstream object detectors. §4.1 describes details on attacking 5 two-stage object detectors and 8 single-stage ones. §4.2 describes the 'targeted' false positives attack as a novel attacking scheme, where the user can specify a desired class be produced by the attacked detector. §4.3 evaluates the transferring ability of the proposed attack method among common network architectures. §4.4 performs three ablation studies on major factors that can affect performance. §4.5 shows the visual examples of attacking performance. (§4.4 and §4.5 are described in Supplementary Material).

## 4.1 Experimental Setup

MS COCO 2014 dataset [17] is used to evaluate the performance of the background patch attack. It contains 80 object class and a background class. We randomly select 2000 images from MS COCO 2014 validation set for experiments. The detection performance is evaluated using "mean average precision" (mAP) metric [8] at Intersection-over-Union (IoU) threshold 0.5 and 0.7.

Experiments are conducted for 5 two-stage object detectors and 8 single-stage ones. For simplicity, we denote the base networks of these object detectors as: `vgg16` **(v16)** [30] , `mobilenet` **(mn)** [12], `resnet50` **(rn50)**, `resnet101` **(rn101)** and `resnet152` **(rn152)** [11]. For two-stage object detectors, we tested Faster-RCNN (**FR**) [29] based on 5 different RPNs: [4] **FR-v16**, **FR-mn**, **FR-rn50**, **FR-rn101** and **FR-rn152**. For single-stage object detectors, we tested **SSD** [19], **YOLO2** [28], **YOLO3** [27], **RFB** [18] and **FSSD**

[4] As the RPNs use the same base network with detectors, we use the name of base network to denote different RPNs

Table 1: *Performance of background patches attack on 5 two-stage object detectors with 5 different Region Proposal Networks (RPNs) and 8 single-stage object detectors at mAP* 0.5 *and* 0.7. **No Noise** *denotes the original performance without adding noise.* **Random** *denotes the performance of adding random noises on patches.* **TPC**, **TPS**, **TPC+TPS**, **FPC** *and* **TPC+TPS+FPC** *denote the performance using corresponding loss terms respectively. Lower value denotes better attacking performance.*

| | No Noise | Random | TPC | TPS | FPC | TPC+TPS+FPC |
|---|---|---|---|---|---|---|
| **FR-v16** | 62.4/48.7 | 62.5/48.9 | 50.7/38.8 | 51.2/38.0 | 50.7/38.3 | **41.9/32.7** |
| **FR-mn** | 46.1/32.9 | 46.4/32.9 | 31.6/22.5 | 34.6/21.2 | 36.0/26.4 | **26.6/19.3** |
| **FR-rn50** | 64.7/52.7 | 64.7/52.2 | 47.7/40.1 | 47.2/35.1 | 52.2/43.9 | **39.8/33.4** |
| **FR-rn101** | 66.0/56.0 | 65.8/55.7 | 39.9/32.4 | 42.0/29.8 | 53.9/48.5 | **36.2/31.2** |
| **FR-rn152** | 70.0/60.0 | 69.1/58.9 | 38.3/30.1 | 42.3/27.2 | 56.0/48.3 | **36.8/31.7** |
| **SSD-rn50** | 46.6/37.2 | 47.2/37.1 | 39.7/30.0 | 37.4/25.4 | 33.9/28.8 | **27.9/20.9** |
| **SSD-v16** | 48.3/37.0 | 47.8/37.1 | 36.9/26.4 | 31.0/18.0 | 26.2/21.9 | **24.5/17.4** |
| **RFB-rn50** | 48.9/40.3 | 48.7/41.2 | 36.2/27.4 | 43.7/38.4 | 31.6/27.7 | **26.1/20.5** |
| **RFB-v16** | 48.3/37.9 | 46.5/37.3 | 32.9/23.8 | 32.3/19.2 | 30.2/25.1 | **26.0/19.4** |
| **YOLO2-mn** | 46.6/30.4 | 45.4/29.9 | 35.7/23.1 | 26.9/15.4 | 23.6/17.5 | **22.3/15.3** |
| **YOLO3-mn** | 49.0/36.0 | 49.6/36.5 | 40.0/28.5 | 33.9/20.7 | 32.7/25.5 | **33.3/21.8** |
| **FSSD-rn50** | 51.2/41.5 | 51.4/42.2 | 39.8/29.7 | 38.3/27.2 | 31.7/31.3 | **28.8/20.8** |
| **FSSD-v16** | 54.0/44.2 | 53.9/43.5 | 38.5/28.4 | 31.8/19.8 | 35.4/31.0 | **33.5/24.1** |

[16] on different base networks: **SSD-v16**, **SSD-rn50**, **RFB-v16**, **RFB-rn50**, **YOLO2-mn**, **YOLO3-mn**, **FSSD-v16** and **FSSD-rn50**.

We evaluated four combinations of loss functions: (1) **TPC**, (2) **TPS**, (3) **FPC** and (4) **TPC+TPS+FPC**. We added a **Random** baseline experiment for comparison, where random noise under normal distribution with the same distortion as in our method is added to the background patches. Table 1 shows the detection performances under attack. Random noise barely affects the performance of object detectors. The performance decreases notably under the adversarial attacks. TPS is less effective in two-stage object detectors than the single-stage ones, as the shape offset attack can be mitigated by the later shape refinement in the sub-network of two-stage detectors. FPC largely reduces the performance of both kinds of detectors, due to the increased false positives. The combined loss of TPC+TPS+FPC achieves the best performance, which decreases the performance by $\sim 42\%$ at mAP 0.5 and $\sim 40\%$ at mAP 0.7 for two-stage object detectors, $\sim 42\%$ at mAP 0.5 and $\sim 47\%$ at mAP 0.7 for single-stage detectors. More visual examples are illustrated in Supplementary Material at §4.5.

## 4.2 Targeted False Positives

The 'targeted' false positives is an important adversarial attack scheme in practice, with the goal to force a detector to mistakenly generate a false positive having specific given class. For example, to corrupt a traffic monitoring detectors to generate false positives of pedestrians in the background, that can greatly weaken the trustworthiness of the detector.

Our method can generate targeted false positives using Eq.(4) with a specified class, *e.g.*, "person". Figure 3 illustrates an example of the "person" targeted false positives attack, where many "person" detections appear in the background. Table 2 shows the performance of the "person" targeted false positives attack on 8 single-stage object detectors, where the detection performance decreases by $\sim 8\%$ at mAP 0.5 and $\sim 9\%$ at mAP 0.7. This indicates that targeted false positive attack is more challenging than an 'untargeted' attack, where the performance largely drops by $\sim 38\%$ at mAP 0.5 and $\sim 32\%$ at mAP 0.7, as shown in Table 1.

Table 2: *Performance of "person" false positives attack on the 8 single-stage object detectors at mAP 0.5 and 0.7.*

| | Targeted FPC |
|---|---|
| **SSD-rn50** | 44.7/35.0 |
| **SSD-v16** | 46.4/35.1 |
| **RFB-rn50** | 45.9/38.8 |
| **RFB-v16** | 43.9/34.2 |
| **YOLO2-mn** | 38.7/26.0 |
| **YOLO3-mn** | 44.8/32.3 |
| **FSSD-rn50** | 48.9/38.9 |
| **FSSD-v16** | 47.5/38.0 |

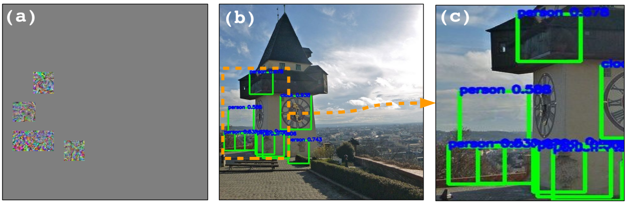

Figure 3: *Visual example of "person" targeted false positives on* **SSD-v16**. *(a) Background patches generated using FPC loss function. (b) Detection result by adding (a). (c) Zoom-in illustration of "person" false positives.*

Table 3: *Performance of transferring attacks between 8 object detectors (4 two-stage and 4 single-stage) at mAP 0.5 and 0.7. The row denotes where the background patches are generated from, and the column denotes each object detectors.*

| | FR-v16 | FR-rn50 | FR-rn101 | FR-rn152 | SSD-v16 | SSD-rn50 | YOLO2-mn | YOLO3-mn |
|---|---|---|---|---|---|---|---|---|
| **No Noise** | 62.4/48.7 | 64.7/52.7 | 66.0/56.0 | 70.0/60.0 | 48.3/37.0 | 46.6/37.2 | 46.6/30.4 | 49.0/36.0 |
| **FR-v16** | **41.9/32.7** | 61.6/49.8 | 63.0/54.3 | 67.8/57.9 | 46.7/35.6 | 46.2/36.7 | 44.5/30.4 | 48.7/35.1 |
| **FR-rn50** | 60.3/47.4 | **39.8/33.4** | 62.0/53.6 | 67.6/57.1 | 47.8/36.0 | 46.4/36.5 | 45.5/30.2 | 48.5/35.2 |
| **FR-rn101** | 62.2/48.0 | 60.7/49.6 | **36.2/31.2** | 66.3/55.4 | 47.7/36.3 | 46.5/36.8 | 47.1/30.8 | 48.3/35.1 |
| **FR-rn152** | 61.9/46.3 | 60.0/48.3 | 59.5/61.0 | **36.8/31.7** | 47.6/36.1 | 46.8/36.8 | 44.8/30.3 | 48.8/35.3 |
| **SSD-v16** | 60.3/48.3 | 64.2/52.3 | 65.1/56.1 | 69.5/59.7 | **24.5/17.4** | 45.4/36.4 | 46.5/30.5 | 48.3/36.2 |
| **SSD-rn50** | 61.4/48.1 | 64.3/52.8 | 65.4/56.2 | 70.4/60.4 | 47.5/35.4 | **27.9/20.9** | 46.5/30.4 | 49.1/36.1 |
| **YOLO2-mn** | 61.6/48.8 | 64.4/52.4 | 65.6/56.7 | 69.7/59.6 | 47.8/35.8 | 46.6/36.9 | **22.3/15.3** | 45.6/32.9 |
| **YOLO3-mn** | 61.7/48.9 | 64.5/51.8 | 64.7/56.3 | 70.0/59.6 | 47.8/36.6 | 46.5/36.0 | 39.9/27.7 | **33.3/21.3** |

### 4.3 Transferring Study

We study the transferring ability (how adversarial attacks generated on one detector can be used to attack another detector), to further explore the properties of vulnerability among common network architectures. Table 3 reports the attack on 4 two-stage detectors and 4 single-stage ones. Our attack can transfer to similar network architectures. Background patches generated from FR-rn50 and FR-152 are more effective to FR-101 than FR-v16. Background patches generated from YOLO3-mn can also be effective to YOLO2-mn. The transferring ability between different base networks are weaker. Also, two-stage and single-stage object detectors can barely transfer attack from each other.

## 5 Conclusion

In this paper, we explore the vulnerability of Single Shot Module (SSM) in mainstream object detectors, by adding imperceptible adversarial perturbations on small background patches outside the object. Our background patches attack can largely decrease the true positives and increase false positives in the background. Experiments on MS COCO 2014 dataset by attacking 5 two-stage object detectors and 8 single-stage ones demonstrate the efficacy. Future work includes the improvement of the optimization process and extension to attack black-box models.

**Acknowledgement.** This material is based upon work supported by, or in part by, the U. S. Army Research Laboratory and the U. S. Army Research Office under contract/grant number 73042CS.

# References

[1] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *arXiv preprint arXiv:1801.00553*, 2018.

[2] Shumeet Baluja and Ian Fischer. Learning to attack: Adversarial transformation networks. In *AAAI*, 2018.

[3] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

[4] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Chau. Robust physical adversarial attack on faster r-cnn object detector. *arXiv preprint arXiv:1804.05810*, 2018.

[5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *IJCV*, 2010.

[6] Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *CVPR*, 2018.

[7] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Physical adversarial examples for object detectors. *arXiv preprint arXiv:1807.07769*, 2018.

[8] Ross Girshick. Fast R-CNN. In *ICCV*, 2015.

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[12] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv 1704.04861*, 2017.

[13] Danny Karmon, Daniel Zoran, and Yoav Goldberg. Lavan: Localized and visible adversarial noise. *arXiv preprint arXiv:1801.02608*, 2018.

[14] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *ICLR*, 2017.

[15] Yuezun Li, Daniel Tian, Mingching Chang, Xiao Bian, and Siwei Lyu. Robust adversarial perturbation on deep proposal-based models. In *BMVC*, 2018.

[16] Zuoxin Li and Fuqiang Zhou. FSSD: Feature fusion single shot multibox detector. *arXiv preprint arXiv:1712.00960*, 2017.

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[18] Songtao Liu, Di Huang, and Yunhong Wang. Receptive field block net for accurate and fast object detection. In *ECCV*, 2018.

[19] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016.

[20] Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Hai Li, and Yiran Chen. Dpatch: An adversarial patch attack on object detectors. *arXiv preprint arXiv:1806.02299*, 2018.

[21] Jiajun Lu, Hussein Sibai, and Evan Fabry. Adversarial examples that fool detectors. *arXiv 1712.02494*, 2017.

[22] Jiajun Lu, Hussein Sibai, Evan Fabry, and David Forsyth. Standard detectors aren't (currently) fooled by physical adversarial stop signs. *arXiv preprint arXiv:1710.03337*, 2017.

[23] Bo Luo, Yannan Liu, Lingxiao Wei, and Qiang Xu. Towards imperceptible and robust adversarial example attacks against neural networks. In *AAAI*, 2018.

[24] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2016.

[25] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *CVPR*, 2017.

[26] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *EuroS&P*, 2016.

[27] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[28] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016.

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *TPAMI*, 2017.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 2014.

[31] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv 1312.6199*, 2013.

[32] Prabhakar Telagarapu, V Jagan Naveen, A Lakshmi Prasanthi, and G Vijaya Santhi. Image compression using DCT and wavelet transformations. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 2011.

[33] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *ICCV*, 2017.

[34] Xiaohui Zeng, Chenxi Liu, Weichao Qiu, Lingxi Xie, Yu-Wing Tai, Chi Keung Tang, and Alan L Yuille. Adversarial attacks beyond the image space. *arXiv 1711.07183*, 2017.