Computer Vision and Image Understanding
journal homepage: www.elsevier.com

TransRPN: Towards the Transferable Adversarial Perturbations Using Region Proposal Networks and Beyond

Yuezun Li[a], Ming-Ching Chang[b], Pu Sun[c], Honggang Qi[c], Junyu Dong[a], Siwei Lyu[d,**]

[a] *Ocean University of China, China*
[b] *University at Albany, State University of New York, USA*
[c] *University of Chinese Academy of Sciences, China*
[d] *University at Buffalo, State University of New York, USA*

ABSTRACT

The adversarial perturbation for object detectors has drawn increasing attention due to the application in video surveillance and autonomous driving. However, few works have explored the transferability of adversarial perturbations across different object detectors. In this paper, we describe a simple but effective method, namely *TransRPN*, to generate adversarial perturbations that can reliably transfer among different object detectors – different categories (*e.g.*, SSD, Faster-RCNN, YOLO) and different base networks (*e.g.*, VGG16, ResNet, MobileNet), and even other tasks such as instance segmentation methods. Our method targets the Region Proposal Network (RPN) as the common bottleneck of the existing object detectors and disrupts the RPN by attacking the intermediate features. Moreover, as RPNs have no constraint on size of input image, our method can generate the adversarial images directly fitting into object detectors with arbitrary input size, which thereby improves the feasibility of our method in practical applications. We study four type of RPNs and validate our method on each type of RPN on MSCOCO dataset with nine object detectors and two instance segmentation methods, as well as the real-world API, which demonstrates the effectiveness of our method regarding the transferability.

1. Introduction

Convolutional Neural Networks (CNNs) have proved to be vulnerable against adversarial perturbations – the intentionally crafted noises that are imperceptible to human observers while can lead CNNs to large errors (Akhtar and Mian, 2018; Chakraborty et al., 2018; Zhang et al., 2020a; Ozbulak et al., 2021). The vulnerability of CNNs to adversarial perturbations suggests that the CNN models do not behave like humans, which can help us to better understand these models and to improve the robustness (Pang et al., 2018; Arnab et al., 2018; Naseer et al., 2020; Stutz et al., 2020) and defending strategies (Tramèr et al., 2017; Mustafa et al., 2020).

The adversarial perturbation was originally proposed in (Szegedy et al., 2014; Goodfellow et al., 2015; Zhou et al., 2018; Brendel et al., 2017) to attack image classifiers with two settings, *i.e.*, white-box attack, where the attackers can access the details of models (Szegedy et al., 2014; Goodfellow et al., 2015), and black-box attack, where the models are unknown to the attackers (Zhou et al., 2018; Brendel et al., 2017). One typical solution for black-box attack is to improve the transferability, which aims to develop strong transferable adversarial perturbations by attacking known models (Dong et al., 2018; Zhou et al., 2018; Wu et al., 2020b; Li et al., 2020b).

**Corresponding author.
e-mail: siweilyu@buffalo.edu (Siwei Lyu)

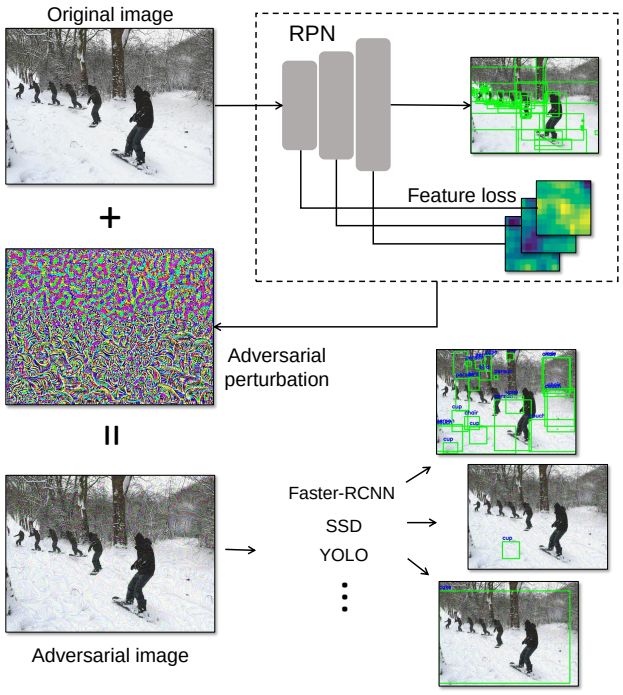

Fig. 1: Overview of TransRPN.

Recently, attacking object detectors has been drawn increasing attentions due to the broadly usage of them in security-critical applications such as video surveillance and autonomous driving (Xie et al., 2017; Chen et al., 2018; Wei et al., 2019; Li et al., 2020a). Compared to image classifiers, object detectors are typically more sophisticated, which output multiple labels and bounding boxes for each object instead of a single image label. Therefore, adversarial perturbation schemes designed for attacking image classifiers cannot be used directly to attack object detectors. To date, most of the existing works (Xie et al., 2017; Chen et al., 2018; Li et al., 2018, 2020a) are designed for the white-box attack on object detectors and a relatively fewer number of them are dedicated to develop transferable adversarial perturbations among different object detectors. The UEA method (Wei et al., 2019) proposed a GAN model (Goodfellow et al., 2014) to generate transferable adversarial perturbations based on Faster-RCNN to disturb SSD. However, this method is only designed for VGG16 (Simonyan and Zisserman, 2014) base network, which is not applicable to object detectors with different base networks. The DR method (Lu et al., 2020) can attack object detectors by transferring adversarial perturbations from image classifiers. However, this method is not dedicated for attacking object detectors, such that the transferability is not fully explored. Moreover, since UEA method utilizes GAN model to generate adversarial images and DR method generates adversarial images based on classifier, they have a common shortcoming is that the size of generated adversarial images is fixed, which can not be directly used to attack object detectors if their input size is different, *e.g.*, an adversarial image generated by GAN model with 300×300 output size can not attack an object detector with 512×512 input size, which evidently hinders the application of transferable adversarial attack in real-world settings.

In this paper, we propose a simple but effective method, namely *TransRPN*, to generate adversarial perturbations that can reliably transfer across different object detectors (*e.g.*, Faster-RCNN (Ren et al., 2017), SSD (Liu et al., 2016)) and YOLO (Redmon et al., 2016; Redmon and Farhadi, 2017, 2018)), and different tasks (*e.g.*, instance segmentation) (Fig. 1). Our method is based on the observation that most of the existing object detectors rely on a Region Proposal Network (RPN) style models at their core – RPN is a major component of Faster-RCNN and also a regression-based model similar to SSD and YOLO. Therefore, different with existing works that attack the whole object detector for transferability, our method works by attacking the RPNs as a common bottleneck of different object detectors. Moreover, since the RPN can take images of arbitrary size, we can directly perturb the original image without the need of resizing as in the cases of (Wei et al., 2019; Lu et al., 2020).

Note our preliminary conference work of (Li et al., 2018) attacks RPN by corrupting the final outputs including confidence score and bounding box regression prediction. In this extension, we design a simple loss function to disturb the intermediate features, the common properties at a feature level, which notably reduces the overfitting phenomenon and improves the transferability. The loss function is then optimized using the momentum iterative fast gradient sign method (Dong et al., 2018). The experiments are conducted on the MSCOCO

dataset (Lin et al., 2014) with nine mainstream object detectors, covering three categories (Faster-RCNN, SSD and YOLO) and five base networks (VGG16 (Simonyan and Zisserman, 2014), ResNet50 (He et al., 2016), ResNet101, ResNet152 and MobileNet (Howard et al., 2017)) and two state-of-the-art instance segmentation methods (Mask-RCNN (He et al., 2017) and YOLACT (Bolya et al., 2019)). Moreover, our method is validated on a real-world API from Facebook named *Detectron2*. These experiments empirically demonstrate the effectiveness of our method. We also investigate the influence of other loss functions and their combinations with different optimization methods on white-box and black-box attacks.

Our contributions are summarized as following:

1. We propose a simple but effective attack method that can reliably transfer among different object detectors by attacking the intermediate features of RPNs.

2. Thanks to the property of RPNs that can take images with arbitrary size, our method can generate adversarial images to directly attack object detectors without changing the size.

3. We comprehensively investigate the transferability of adversarial perturbations based on RPNs under different settings that previous methods do not consider.

4. The experiments are conducted on various object detectors, and instance segmentation methods, as well as the real-world API, to demonstrate the efficacy of our method on transferability.

This paper extends the preliminary conference paper (Li et al., 2018) in several aspects: 1) We propose TransRPN, which extends the task-specific loss functions, the confidence loss and shape loss, which reduce confidence score and disturb the bounding box shape regression of correct object proposals, to intermediate feature map loss so as to improve the transferability of adversarial perturbations. 2) We study the performance of proposed method towards various object detectors covering proposal-based and regression-based categories, as well as instance segmentation methods and real-world API. 3) We study the robustness of proposed methods towards adversarial defense strategies and image compression. 4) We conduct ablation study on the effect of different loss combinations and other strategies regarding the transferability.

2. Related Works

2.1. Adversarial Perturbations

Adversarial perturbations are intentionally crafted noises that can cause the CNN models to make mistakes. The adversarial perturbations are first designed to disrupt image classifiers (Szegedy et al., 2014; Goodfellow et al., 2015; Kurakin et al., 2017; Papernot et al., 2016b; Moosavi-Dezfooli et al., 2016, 2017; Zeng et al., 2017; Luo et al., 2018; Baluja and Fischer, 2018; Poursaeed et al., 2018). Specifically, many works focus on white-box attack by using the model gradient to change the input images across the decision boundary (Szegedy et al., 2014; Goodfellow et al., 2015; Kurakin et al., 2017; Dong et al., 2018), while another vein focuses on improving the transferability of adversarial perturbations (Papernot et al., 2016a; Liu et al., 2017; Mopuri et al., 2018; Dong et al., 2018, 2019; Zhou et al., 2018; Li et al., 2020c; Huan et al., 2020; Zhou et al., 2020; Lu et al., 2020; Huang and Zhang, 2020; Wu et al., 2020a).

Recently, adversarial perturbations are extended to object detectors (Xie et al., 2017; Chen et al., 2018; Li et al., 2018, 2019; Wei et al., 2019; Wang et al., 2020; Zhang et al., 2020b; Li et al., 2020a; Chow et al., 2020a,b; Wu et al., 2020c; Serban et al., 2020). The work in (Xie et al., 2017) first explored the weakness of Faster-RCNN object detector using Dense Adversary Generation (DAG) method in the white-box setting. The work in (Wang et al., 2020) extends DAG with Projected Gradient Descent (PGD) method to improve efficiency. The work in (Chow et al., 2020a) described a Targeted Adversarial Objectness Gradient Attacks on real-time object detectors. Then the response of object detectors under different attack methods was studied in work (Chow et al., 2020b). Another work in (Zhang et al., 2020b) disrupted the contextual information of objects to further disturb the predictions and the work of (Li et al., 2020a) focused on finding the universal perturbation pattern for target

models. However, few works focus on improving the transferability among different object detectors, especially different categories. The UEA method (Wei et al., 2019) utilized a GAN (Goodfellow et al., 2014) model to generate transferable adversarial perturbations among object detectors. However, this method is only validated on VGG16 based Faster-RCNN and SSD. The work (Lu et al., 2020) studied the transferability of adversarial perturbation from attacking image classifier to other tasks such as object detectors. In addition, the existing methods always generate adversarial images with fixed size due to their internal mechanism, which is not well-suited in practical use. In this work, we develop a simple method to create transferable adversarial perturbation by attacking RPNs, which can notably affect the performance of different object detectors and other tasks such as instance segmentation.

2.2. Object Detectors and Region Proposal Networks

The recent mainstream object detectors can be divided into three categories: Faster-RCNN (Ren et al., 2017), SSD (Liu et al., 2016) and YOLO (Redmon et al., 2016; Redmon and Farhadi, 2017, 2018). The Faster-RCNN is proposal-based architecture that begins with a Region Proposal Network (RPN) to generate object proposals, which are then forwarded to a subnetwork for refinement. The SSD and YOLO are regression-based architecture where the final object detection can be obtained in a single forward pass. The major difference between them is SSD employs multiple feature maps to predict results, while YOLO uses the last feature map for prediction. Since the architecture of these object detectors is very different, the transferability of directly using one's adversarial perturbation to attack others is weak (Xie et al., 2017; Li et al., 2018; Wei et al., 2019).

The Region Proposal Networks (RPNs) is a regression-based architecture, which outputs a set of object proposals in the form of bounding boxes in a single forward pass. Concretely, a set of predefined bounding boxes (anchor boxes) is initialized, then the confidence score and location adjustment of each anchor box are predicted. Due to its efficiency and accuracy, RPNs are widely used in various tasks to provide object proposals. As the RPN is a core component of Faster-RCNN and similar to other regression-based models of SSD and YOLO, it can share some common properties among the different object detectors. Therefore, we target the RPN as the common bottleneck of different object detectors.

3. Methodology

In this section, we first introduce the task-specific loss terms proposed in preliminary conference paper (Li et al., 2018) – the confidence loss and shape loss, which correspond to disturb the confidence score and the shape regression of correct object proposals respectively. Then we introduce a new task-agnostic loss used in TransRPN, the feature loss, which can greatly improve the attack transferability by disturbing the intermediate feature maps.

3.1. Notations and Formulation

Let $\mathcal{I}$ denote the benign image. We denote $\mathcal{F}_\theta$ as the mapping function of the Region Proposal Network (RPN) with parameters θ. $\{\bar{b}_i = (\bar{x}_i, \bar{y}_i, \bar{w}_i, \bar{h}_i)\}_{i=1}^{n}$ denotes 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. Denote $\mathcal{I}^{'}$ as the adversarial image. Let $\{(s_j^{'}, b_j^{'})\}_{j=1}^{m} = \mathcal{F}_\theta(\mathcal{I}^{'})$ denote the results of RPN on the adversarial image $\mathcal{I}^{'}$, where $s_j^{'}$ denotes the confidence score, $b_j^{'}$ denotes the bounding box of the j-th object proposal, and m is the number of object proposals. 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. Denote the intermediate feature set of RPN on benign image $\mathcal{I}$ and adversarial image $\mathcal{I}^{'}$ as $\{f_i\}_{i=1}^{k}$ and $\{f_i^{'}\}_{i=1}^{k}$, where $f_i, f_i^{'}$ denote a feature map and k is the number of feature maps . Our goal is to seek an adversarial image $\mathcal{I}^{'}$ that can disturb the results of RPN, while retains imperceptible distortion compared to the benign image $\mathcal{I}$. Thus it can be defined as an optimization problem of designed loss function L in a form of

$$\min L(\mathcal{I}^{'}; \theta, \mathcal{I}), \text{ s.t. } \|\mathcal{I}^{'} - \mathcal{I}\|_\infty \leq \epsilon. \tag{1}$$

Following the works (Goodfellow et al., 2015; Xie et al., 2019), we use ℓ_∞ norm to measure the distortion of adversarial perturbations. ϵ is the budget of adversarial perturbation distortion.

3.2. Revisiting Task-specific Loss

The task-specific losses aim to disturb the final results of RPN. We propose two loss terms – confidence loss and shape loss, to disturb the confidence score and bounding box regression of object proposals respectively.

Confidence Loss. Note the larger the confidence score, the generated proposal is more like an object. Thus this loss aims to reduce the confidence score of correct object proposals, such that they can not be selected in final results. To be more effective, we only attack a set of proposal candidates that are potentially correct. We use $z_j = 1$ to denote the proposal b_j is potentially correct if the IoU overlapping of b_j with its ground truth box is greater than threshold 0.5, $z_j = 0$ otherwise. Thus the confidence loss can be defined as

$$L_c(\mathcal{I}^{'}; \theta, \mathcal{I}) = \sum_{j=1}^{m} z_j \log(s_j^{'}). \tag{2}$$

Minimizing Eq.(2) decreases the confidence score of selected object proposals.

Shape Loss. Besides confidence score prediction, bounding box regression is also an important step to refine the localization of proposals, where the locations of anchor (predefined) boxes are adjusted to match the corresponding ground truth boxes. Therefore, we design a shape loss to explicitly disturb the bounding box shape regression, such that the correct object proposals will be pushed away from their desired locations. 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. Similar to confidence loss, we only consider the set of potentially correct object proposals. Thus the shape loss can be defined as

$$\begin{aligned} L_s(\mathcal{I}^{'}; \theta, \mathcal{I}) = \exp\{-\sum_{j=1}^{m} z_j \cdot \\ \Big[(\Delta x_j^{'} - \Delta \bar{x}_j)^2 + (\Delta y_j^{'} - \Delta \bar{y}_j)^2 \\ +(\Delta w_j^{'} - \Delta \bar{w}_j)^2 + (\Delta h_j^{'} - \Delta \bar{h}_j)^2\Big]\}. \end{aligned} \tag{3}$$

Minimizing Eq.(3) encourages pushing the predicted offsets away from the true offsets

3.3. TransRPN

In contrast to the task-specific loss in preliminary conference paper, we propose a task-agnostic loss – the feature loss to disturb the intermediate feature maps of RPN for improving the transferability among different object detectors. To do so, we increase the difference between the feature maps of adversarial image and benign image. Thus the feature loss can be formulated as

$$L_f(\mathcal{I}^{'}; \theta, \mathcal{I}) = \frac{1}{k} \cdot \sum_{i=1}^{k} \frac{f_i^T \cdot f_i^{'}}{\|f_i\| \cdot \|f_i^{'}\|} \tag{4}$$

We use cosine distance to measure the similarity of features $\{f_i\}_{i=1}^k$ and $\{f_i^{'}\}_{i=1}^k$ as it can normalize the distance to a range $[-1, 1]$. Minimizing the Eq.(4) can enlarge the errors between two sets.

3.4. Optimization

We use the feature loss of Eq.(4) in TransRPN for example. We optimize the loss function using the iterative gradient descent scheme. Inspired by (Dong et al., 2018), we use the sign of gradient with the momentum at each iteration to maintain the running efficiency and efficacy. Let t denote iteration number and $\mathcal{I}_t^{'}$ denote the adversarial image at iteration t. The initial image is set as $\mathcal{I}_0^{'} = \mathcal{I}$. The gradient at iteration $t+1$ can be calculated as

$$g_{t+1} = \lambda \cdot g_t + \frac{\nabla_{\mathcal{I}_t^{'}}(L_f(\mathcal{I}_t^{'}; \theta, \mathcal{I}))}{\|\nabla_{\mathcal{I}_t^{'}}(L_f(\mathcal{I}_t^{'}; \theta, \mathcal{I}))\|_1}, \tag{5}$$

where $\nabla_{\mathcal{I}_t^{'}}$ denotes the gradient of loss function with respect to the input image $\mathcal{I}_t^{'}$ at iteration t, λ is the decay factor of momentum and g_t is accumulated gradient with the momentum. Then the adversarial image $\mathcal{I}_{t+1}^{'}$ can be updated as

$$\mathcal{I}_{t+1}^{'} = \text{clip}\{\mathcal{I}_t^{'} - \alpha \cdot \text{sign}(g_{t+1})\}, \tag{6}$$

where "sign" takes the sign of gradient and α is the step size, "clip" puts the pixel value of image $\mathcal{I}_{t+1}^{'}$ to $[0, 255]$. The process is repeated until (1) the maximum iteration number T

Algorithm 1 *Overview of TransRPN generation.*

Input: Region proposal network $\mathcal{F}_\theta$; benign image $\mathcal{I}$; maximum iteration number T; Distortion budget ϵ

$\mathcal{I}^{'}_0 = \mathcal{I}, t = 0$

while $t < T$ **do**

  $g_{t+1} = \lambda \cdot g_t + \frac{\nabla_{\mathcal{I}^{'}_t}(L_f(\mathcal{I}^{'}_t;\theta,\mathcal{I}))}{\|\nabla_{\mathcal{I}^{'}_t}(L_f(\mathcal{I}^{'}_t;\theta,\mathcal{I}))\|_1}$;

  $\mathcal{I}^{'}_{t+1} = \text{clip}\{\mathcal{I}^{'}_t - \alpha \cdot \text{sign}(g_{t+1})\}$

  if $\|\mathcal{I}^{'}_{t+1} - \mathcal{I}\| > \epsilon$ **then**

    Break

  $t = t + 1$

Output: Adversarial image $\mathcal{I}^{'}_t$

is reached, or (2) the budget of adversarial perturbation ϵ is reached. The overview of TransRPN generation is shown in Algorithm 1.

4. Experiments

In this section, we focus on validating the effectiveness of TransRPN towards transferability with several main-steam object detectors, instance segmentation methods as well as real-world APIs. We also conduct ablation study on the effect of the confidence loss, shape loss, and their combinations with the feature loss as well as other strategies. We then study the robustness against defence strategies.

4.1. Experimental Settings

4.1.1. Dataset

We evaluate the performance of our method on the MSCOCO dataset (Lin et al., 2014). In our experiments, we randomly select 1000 images from the validation set. The detection performance is evaluated using "mean average precision" (mAP) metric (Everingham et al., 2010) at Intersection-over-Union (IoU) threshold 0.5. The range of mAP is $[0, 1]$, where less score denotes worse detection performance, *i.e.*, better attacking performance.

4.1.2. Networks

We investigate four RPNs, which are based on VGG16 (RPN-vgg16), ResNet50 (RPN-res50), ResNet101 (RPN-res101) and ResNet152 (RPN-res152) respectively. Then we attack three type of object detectors, Faster-RCNN, SSD and YOLO. Each type of object detectors includes several variants based on different base networks. For Faster-RCNN, we study four variants with VGG16 (FR-vgg16), ResNet50 (FR-res50), ResNet101 (FR-res101) and ResNet152 (FR-res152) as base networks respectively. For SSD, we study three variants based on VGG16 (SSD-vgg16) , ResNet50 (SSD-res50) and MobileNet (SSD-mn). For YOLO, we study two versions: YOLOv2 and YOLOv3 based on Mobilenet (YOLOv2-mn, YOLOv3-mn).

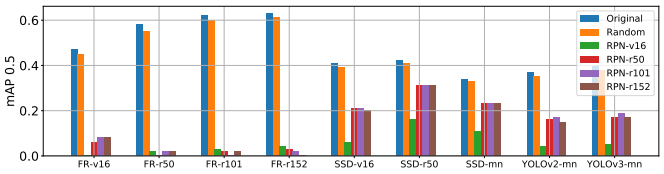

Fig. 2: Illustration of the performance of TransRPN on different object detectors. Each group denotes the performance of different TransRPN attack to corresponding object detector.

Moreover, our method is validated on two state-of-the-art instance segmentation methods: Mask-RCNN and YOLACT. The Mask-RCNN is a proposal-based method which is firstly based on RPN to provide object proposals. The YOLACT is a real-time method which does not reply on RPN and utilize the RetinaNet (Lin et al., 2017b) as base network.

4.1.3. Implementation Details

We select three feature maps ($k = 3$) over each type of RPN to calculate the loss. For RPN-vgg16, we select the feature maps after Conv3-3, Conv4-3 and Conv5-3. For RPN-res50, RPN-r101 and RPN-r152, we select the feature maps after first three residual blocks.

Our experiments are conducted using PyTorch (Paszke et al., 2019) on Ubuntu 16.04 with one Nvidia GPU TITANX. Following works (Luo et al., 2015; Xie et al., 2019), the perturbation budget is set as $\epsilon = 15$. Other parameters are set as $\lambda = 0.5, \alpha = 1, T = 20$.

Table 1: Performance of our method on attacking different object detectors on MSCOCO dataset. "Original" denotes no perturbations added on input image. "Random" denotes random noises added on input image.

Attacks		FR-v16	FR-r50	FR-r101	FR-r152	SSD-v16	SSD-r50	SSD-mn	YOLOv2-mn	YOLOv3-mn
Original		0.47	0.58	0.62	0.63	0.41	0.42	0.34	0.37	0.40
Random		0.45	0.55	0.60	0.61	0.39	0.41	0.33	0.35	0.38
TransRPN	RPN-v16	**0.00**	0.02	0.03	0.04	**0.06**	**0.16**	**0.11**	**0.04**	**0.05**
	RPN-r50	0.06	**0.00**	0.02	0.03	0.21	0.31	0.23	0.16	0.17
	RPN-r101	0.08	0.02	**0.00**	0.02	0.21	0.31	0.23	0.17	0.19
	RPN-r152	0.08	0.02	0.02	**0.00**	0.20	0.31	0.23	0.15	0.17

4.2. Performance on Object Detectors

The performance of our method is shown in Table 1. The leftmost column denotes the attacking methods. The top row denotes different object detectors. "Original" denotes no perturbations added on input image. "Random" denotes random noises with same distortion budget in our method added on input image. The results reveal the randomly added noises merely have effect to object detectors, yet our method can notably degrade the performance of all object detectors. Since the RPN is the core component of Faster-RCNN, our method can greatly reduce their mAP scores, especially for the ones with same base networks as in PRNs, *e.g.*, the mAP score is reduced to approximate zero in FR-v16 and FR-50 based on RPN-v16 and FR-50 respectively. For SSD and YOLO, the performance is also reduced notably as the SSD and YOLO share common properties with PRNs even though the base networks are different. Another observation is the performance of TransRPN is different in terms of different RPNs. The TransPRN on RPN-v16 has the best transferability compared to the other RPNs. It is probably due to the complex structure in ResNet such as Residual block and skip connection may reduce the generalization of gradient compared to VGG16. The performance comparison of our method on different object detectors is shown in Fig. 2. Each group denotes the performance of different TransRPN attack to corresponding object detector. Fig. 3 illustrates several examples of our method based on RPN-v16 on attacking different object detectors, where most objects are mis-detected and several false detections are raised.

Table 2: The performance of our method compared with UEA on VOC07 dataset. Note the Faster-RCNN and SSD300 are both VGG16 based as in UEA. The image size is 300×300.

Attacks	Faster-RCNN	SSD300	Iterations
Original	0.70	0.68	-
DAG	0.05	0.64	$150 \sim 200$
UEA	0.05	**0.20**	-
TransRPN	**0.04**	0.26	≤ 20

4.2.1. Comparisons with the State-of-the-art Methods

We compare our TransRPN based on RPN-vgg16 with three state-of-the-art methods: DAG (Xie et al., 2017), UEA (Wei et al., 2019) and DR (Lu et al., 2020). Despite DAG tests the transferability among object detectors, it focuses on white-box attacks to object detectors. The UEA is dedicated to transfer the adversarial perturbation across Faster-RCNN and SSD on VGG16 base network using a GAN. To fairly compare with UEA, our method is tested in the same setting as described in UEA (Wei et al., 2019). Specifically, the Faster-RCNN and SSD object detectors in UEA are based on VGG16 and trained on VOC0712 dataset (Everingham et al., 2010) using the implementation *Simple Faster-RCNN*[1] and *Torch-SSD300*[2]. Since UEA requires to construct and train an extra GAN model to generate adversarial perturbation, the image size has to be fixed. We follow UEA to set the input size as 300×300 and perform our method to attack the same implementation of Faster-RCNN

[1] https://github.com/chenyuntc/simple-faster-rcnn-pytorch
[2] https://github.com/kuangliu/torchcv

Fig. 3: Visual examples of our method based on RPN-v16 on different object detectors.

Table 3: The performance of our method compared with DR method on MSCOCO dataset. The image size is 224×224.

Attacks	FR-v16	FR-r50	FR-r101	FR-r152	SSD-v16	SSD-r50	SSD-mn	YOLOv2-mn	YOLOv3-mn	Iterations
Original	0.22	0.28	0.28	0.28	0.23	0.24	0.22	0.18	0.18	-
Random	0.22	0.26	0.26	0.28	0.21	0.21	0.19	0.16	0.16	-
DR	0.12	0.17	0.18	0.19	0.08	0.10	0.08	0.06	0.06	1000
TransRPN	**0.00**	**0.02**	**0.03**	**0.03**	**0.04**	**0.05**	**0.03**	**0.03**	**0.03**	$\leq$ **20**

and SSD, then we test our method using VOC07 testing set as in UEA. The results of TransRPN and other methods including the iteration numbers are shown in Table 2. The DAG method takes 150 $\sim$ 200 iterations (referred from (Wei et al., 2019)) while our method takes less than 20 iterations and has much better performance. Compared to UEA, our method is quite simple and can also achieve the competitive performance on both Faster-RCNN and SSD. More importantly, our method can be effective on more different object detectors, see Table 1.

The DR method transfers the adversarial perturbation from ImageNet image classifier to object detectors. Thus the image size has to be fixed as 224×224. To fairly compare with DR, we perform our method with same input size as DR. Since the DR method has released the code[3], we apply this method to the MSCOCO dataset with nine object detectors used in our experiment for comparison. As shown in Table 3, our method outperforms DR in all object detectors. Moreover, the DR method requires 1000 iterations per image, which is notably slower than our method.

Therefore, compared to the existing methods, our method

[3] https://github.com/erbloo/dr_cvpr20

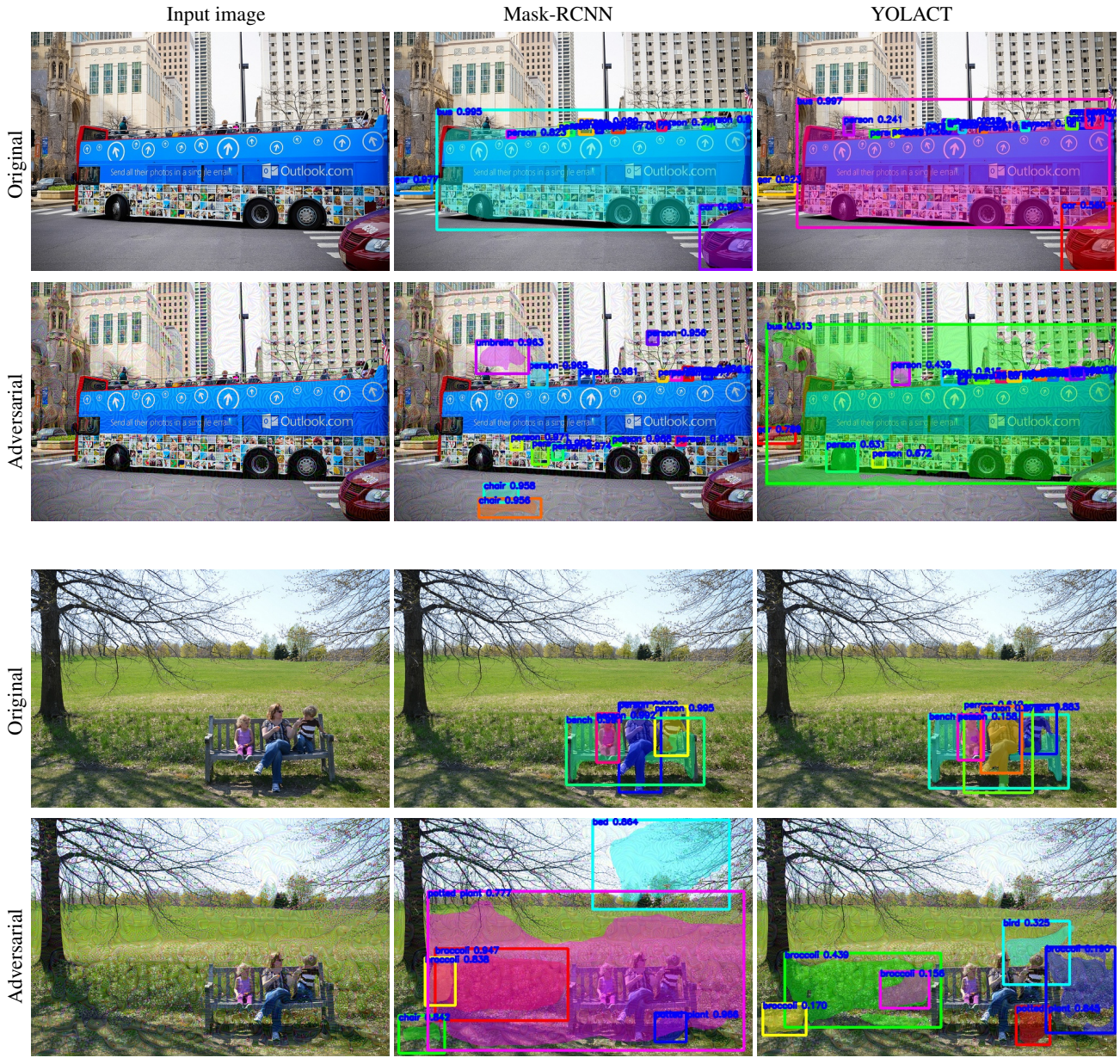

Fig. 4: Visual examples of our method based on RPN-v16 on different instance segmentation methods.

(1) achieves competitive even better performance in their respect settings, (2) can directly attack the original image without changing the size, (3) is effective on more different object detectors.

4.3. Ablation Studies

4.3.1. Different attack settings

To further understand the transferability of RPNs, we investigate the performance of different attack settings regarding the white-box and black-box attacks. Specifically, we study the impact of different loss functions and their combinations, as well as the strategies for improving transferability such as input transformation and optimization schemes mixture in DIM (Xie et al., 2019).

- *Different Loss Functions.* We study the effect of confidence loss, shape loss and feature loss, as well as their the combinations.

- *Input Transformation.* The DIM method discovered the image transformation such as random resizing and

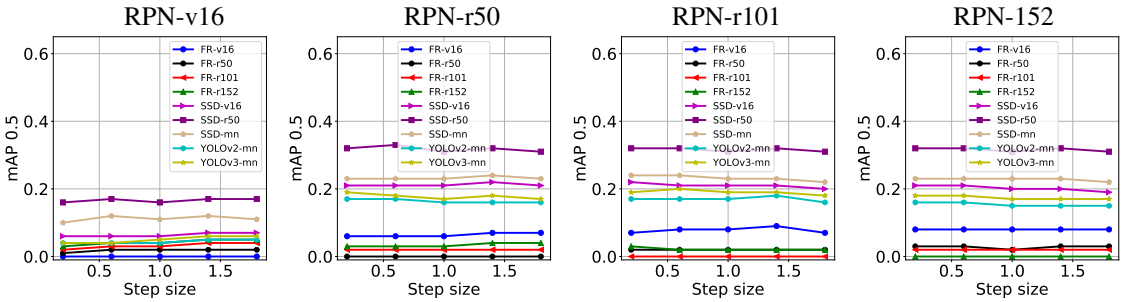

Fig. 5: Ablation study of our method regarding step size.

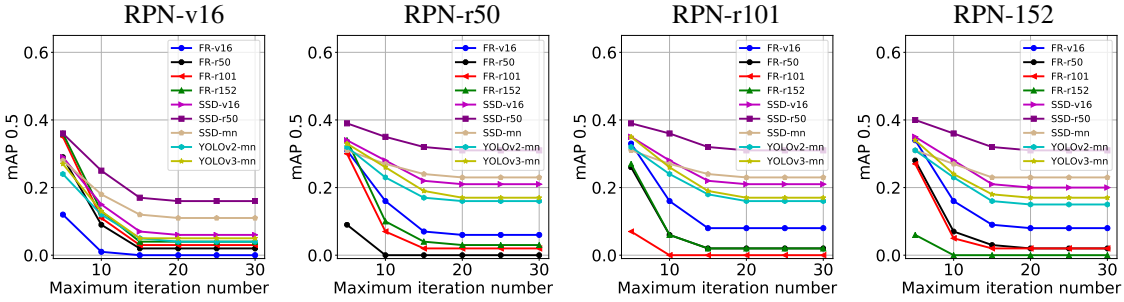

Fig. 6: Ablation study of our method regarding iteration number.

padding with zero can improve the transferability of adversarial perturbations on attacking image classifiers. Thus we apply this strategy to our method and observe the effect to the transferability among different object detectors.

- *Optimization Schemes Mixture.* The DIM method also observed the mixture of different optimization schemes can improve the transferability on attacking image classifiers. Thus we adapt the iterative fast gradient sign method (Goodfellow et al., 2015) together with our method to attack object detectors.

For simplicity, we denote the confidence loss as L_c, shape loss as L_s and the feature loss used in our method as L_f. We use Υ to denote whether input transformation is applied and $\mathcal{M}$ to denote whether mixture of multiple optimization scheme is applied. Thus different attack settings can be represented by the combination of these symbols. For example, $\{L_s, L_f, \Upsilon\}$ denotes the loss function is composed by confidence loss and feature loss, and the input transformation is applied. The ablation study of our method on all RPNs is shown in Table 4. The results reveal the transferability of using confidence loss or shape loss or their combination is weaker than solely using the feature loss. It is because the confidence loss or shape loss target the final prediction, which is likely to overfit to the RPN architecture. Moreover, combining confidence loss or shape loss with feature loss can also reduce the transferability, as the combination distracts a portion of gradient. We also discover the strategies effect in attacking image classifiers such as input transformation and mixture of multiple optimization scheme have merely effect on improving the transferability among the object detectors, which is probably due to the mechanism of object detection is more sophisticated than image classifiers.

4.3.2. Step Size

We investigate the influence of the step size to our method. We set the step size in range $[0.2, 2]$ and disable the limitation of maximum iteration number. The performance of our method on attacking nine object detectors is shown in Fig.5. We can observe the mAP score is stable with step size changing, which indicates our method is not sensitive to the step size.

4.3.3. Maximum Iteration Number

We then study the impact of iteration number to our method. The iteration number is set in range $[5, 30]$. The Fig.6 illustrates

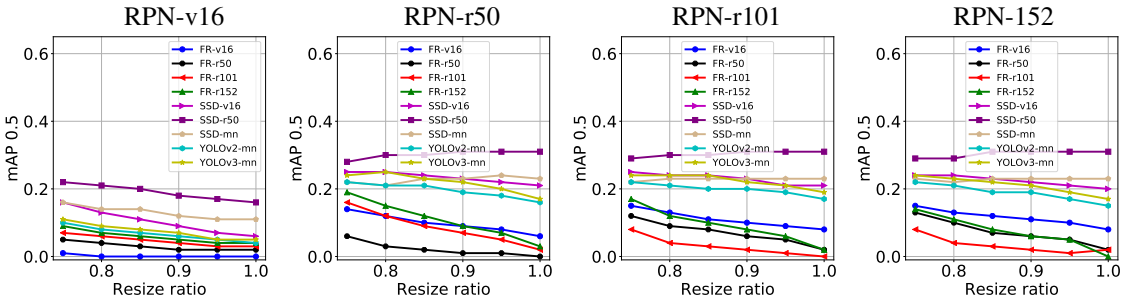

Fig. 7: The performance of our method on adversarial images towards adversarial defense.

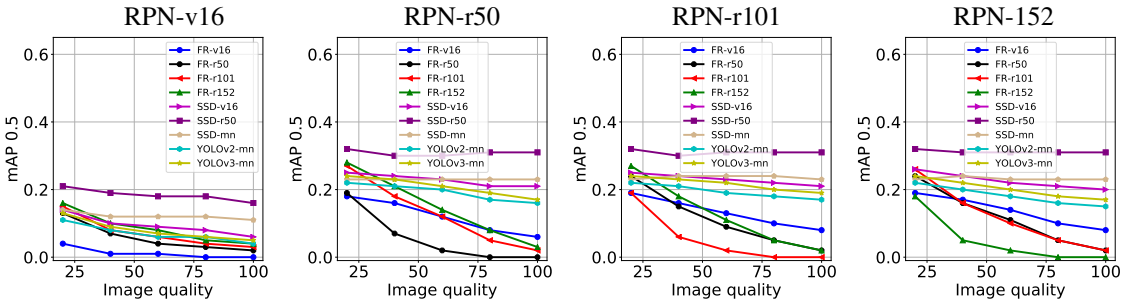

Fig. 8: The performance of our method on adversarial images towards image compression.

the performance trend of our method on nine object detectors with maximum iteration number increasing. This figure reveals the mAP score decreases as the maximum iteration number increasing, but the curve becomes flat after approximate 20 iteration number as more iterations does not improve extra attacking performance as the budget of adversarial perturbation has been reached.

4.4. Robustness

We study the robustness of our method under two scenarios: Adversarial defense and image compression.

4.4.1. Adversarial Defense

We investigate the robustness of our method with regards to the adversarial defense[4]. Note the existing works to defend the adversarial perturbations are dedicated to image classifiers such as Defense-GAN (Samangouei et al., 2018), HPG (Liao et al., 2018), which are generation-based such that they are not trivial to directly be applied to object detectors. Therefore, we adapt the image transformation based method proposed in (Xie et al., 2018), which applies random resizing and padding around with zero to mitigate the adversarial effect, to our task. Specifically, we set the resizing ratio varying from $[0.75, 1]$ and see the response of our method. The result of our method against the defense is shown in Fig. 7. We can observe the detection performance only slightly increases with the resize ratio reducing.

4.4.2. Image Compression

We study the robustness of our method with regards to image compression. Specifically, we change the quality of images from 20 to 100 using OpenCV tool, where the larger value denotes the higher image quality, the 100 denotes no compression is applied. Fig. 8 shows the performance of our method with regards to the image compression. Our method reveals the similar trend on all RPNs that the mAP performance only slightly increases as the image quality decreases, *e.g.*, the mAP score at image quality 20 is less than 0.25 on RPN-v16.

4.5. Discussion

Table 1 reveals that our method based on RPN-v16 has better transferability performance compared to the methods based on

[4] Adversarial defense is the strategy that can mitigate the effect of adversarial perturbations.

Table 4: Ablation study of different attack settings based on RPN-v16, RPN-r50, RPN-r101 and RPN-r152 respectively. Note $\{L_f\}$ denotes the setting used in TransRPN.

RPN-v16

Attacks	FR-v16	FR-r50	FR-r101	FR-r152	SSD-v16	SSD-r50	SSD-mn	YOLOv2-mn	YOLOv3-mn
$\{L_c\}$	0.02	0.36	0.42	0.43	0.34	0.39	0.31	0.32	0.34
$\{L_s\}$	0.02	0.22	0.28	0.32	0.29	0.36	0.27	0.26	0.27
$\{L_s, L_f\}$	0.00	0.02	0.04	0.05	0.09	0.22	0.15	0.06	0.07
$\{L_c, L_f\}$	0.00	0.08	0.10	0.11	0.14	0.27	0.20	0.13	0.15
$\{L_c, L_s\}$	0.02	0.36	0.42	0.44	0.34	0.39	0.32	0.32	0.34
$\{L_f, \Upsilon\}$	0.03	0.11	0.12	0.13	0.11	0.20	0.14	0.10	0.11
$\{L_f, \mathcal{M}, \Upsilon\}$	0.04	0.10	0.12	0.13	0.10	0.20	0.14	0.09	0.10
$\{L_s, L_f, \mathcal{M}, \Upsilon\}$	0.03	0.08	0.10	0.11	0.10	0.22	0.14	0.08	0.09
$\{L_c, L_f, \mathcal{M}, \Upsilon\}$	0.02	0.12	0.14	0.15	0.13	0.25	0.19	0.13	0.14
$\{L_c, L_s, L_f, \mathcal{M}, \Upsilon\}$	0.02	0.12	0.14	0.15	0.14	0.26	0.18	0.13	0.14
$\{L_f\}$	**0.00**	**0.02**	**0.03**	**0.04**	**0.06**	**0.16**	**0.11**	**0.04**	**0.05**

RPN-r50

Attacks	FR-v16	FR-r50	FR-r101	FR-r152	SSD-v16	SSD-r50	SSD-mn	YOLOv2-mn	YOLOv3-mn
$\{L_c\}$	0.40	0.05	0.46	0.49	0.39	0.41	0.33	0.35	0.38
$\{L_s\}$	0.38	0.01	0.34	0.41	0.37	0.41	0.32	0.33	0.36
$\{L_s, L_f\}$	0.18	0.00	0.07	0.10	0.30	0.37	0.29	0.25	0.27
$\{L_c, L_f\}$	0.22	0.00	0.14	0.17	0.31	0.38	0.30	0.29	0.32
$\{L_c, L_s\}$	0.41	0.05	0.46	0.49	0.38	0.41	0.33	0.35	0.38
$\{L_f, \Upsilon\}$	0.15	0.05	0.11	0.13	0.23	0.32	0.24	0.19	0.21
$\{L_f, \mathcal{M}, \Upsilon\}$	0.14	0.05	0.10	0.12	0.21	0.30	0.23	0.17	0.19
$\{L_s, L_f, \mathcal{M}, \Upsilon\}$	0.19	0.05	0.13	0.16	0.25	0.33	0.25	0.21	0.23
$\{L_c, L_f, \mathcal{M}, \Upsilon\}$	0.22	0.04	0.18	0.20	0.28	0.35	0.27	0.25	0.27
$\{L_c, L_s, L_f, \mathcal{M}, \Upsilon\}$	0.21	0.04	0.18	0.20	0.28	0.35	0.27	0.24	0.28
$\{L_f\}$	**0.06**	**0.00**	**0.02**	**0.03**	**0.21**	**0.31**	**0.23**	**0.16**	**0.17**

RPN-r101

Attacks	FR-v16	FR-r50	FR-r101	FR-r152	SSD-v16	SSD-r50	SSD-mn	YOLOv2-mn	YOLOv3-mn
$\{L_c\}$	0.40	0.40	0.03	0.41	0.38	0.41	0.33	0.34	0.38
$\{L_s\}$	0.38	0.31	0.01	0.36	0.37	0.41	0.33	0.34	0.36
$\{L_s, L_f\}$	0.22	0.09	0.00	0.09	0.31	0.37	0.30	0.27	0.29
$\{L_c, L_f\}$	0.33	0.23	0.00	0.23	0.35	0.40	0.32	0.33	0.35
$\{L_c, L_s\}$	0.40	0.39	0.02	0.41	0.38	0.41	0.33	0.35	0.38
$\{L_f, \Upsilon\}$	0.17	0.11	0.06	0.12	0.23	0.32	0.25	0.19	0.22
$\{L_f, \mathcal{M}, \Upsilon\}$	0.15	0.10	0.05	0.10	0.22	0.31	0.23	0.18	0.21
$\{L_s, L_f, \mathcal{M}, \Upsilon\}$	0.20	0.13	0.05	0.13	0.27	0.34	0.26	0.23	0.25
$\{L_c, L_f, \mathcal{M}, \Upsilon\}$	0.28	0.22	0.06	0.22	0.31	0.37	0.29	0.28	0.30
$\{L_c, L_s, L_f, \mathcal{M}, \Upsilon\}$	0.30	0.24	0.06	0.24	0.31	0.36	0.29	0.28	0.31
$\{L_f\}$	**0.08**	**0.02**	**0.00**	**0.02**	**0.21**	**0.31**	**0.23**	**0.17**	**0.19**

RPN-r152

Attacks	FR-v16	FR-r50	FR-r101	FR-r152	SSD-v16	SSD-r50	SSD-mn	YOLOv2-mn	YOLOv3-mn
$\{L_c\}$	0.39	0.41	0.41	0.02	0.38	0.41	0.34	0.35	0.37
$\{L_s\}$	0.38	0.34	0.32	0.02	0.38	0.41	0.33	0.34	0.36
$\{L_s, L_f\}$	0.23	0.12	0.09	0.00	0.31	0.38	0.29	0.27	0.28
$\{L_c, L_f\}$	0.34	0.29	0.25	0.00	0.36	0.40	0.32	0.32	0.35
$\{L_c, L_s\}$	0.40	0.41	0.40	0.02	0.38	0.41	0.34	0.34	0.37
$\{L_f, \Upsilon\}$	0.17	0.13	0.12	0.06	0.22	0.31	0.23	0.18	0.21
$\{L_f, \mathcal{M}, \Upsilon\}$	0.16	0.12	0.11	0.06	0.22	0.31	0.23	0.18	0.21
$\{L_s, L_f, \mathcal{M}, \Upsilon\}$	0.23	0.18	0.15	0.07	0.28	0.35	0.26	0.24	0.25
$\{L_c, L_f, \mathcal{M}, \Upsilon\}$	0.30	0.27	0.23	0.07	0.32	0.37	0.29	0.28	0.31
$\{L_c, L_s, L_f, \mathcal{M}, \Upsilon\}$	0.30	0.27	0.24	0.07	0.32	0.37	0.29	0.28	0.31
$\{L_f\}$	**0.08**	**0.02**	**0.02**	**0.00**	**0.20**	**0.31**	**0.23**	**0.15**	**0.17**

RPN-r50, RPN-r101 and RPN-r152. For example, TransRPN on RPN-v16 can reduce the mAP score of YOLOv2-mn from 0.37 to 0.04, while others can only reduce the mAP score to $0.16, 0.17, 0.15$ respectively. Furthermore, Fig. 7 and Fig. 8 shows our method on RPN-v16 is more robust against adversarial defense and image compression than others. These results indicate that the performance of transferable adversarial attack probably relates to network architecture, and complex networks such as ResNet may have weaker transferability than simple ones such as VGG16.

Table 5: The performance of our method on attacking two state-of-the-art instance segmentation methods.

Attacks	Mask-RCNN	YOLACT
Original	0.54	0.49
TransRPN	0.02	0.07

4.6. Performance on Instance Segmentation

We also use our method to attack two state-of-the-art instance segmentation methods: Mask-RCNN and YOLACT. We directly attack these two methods using the adversarial images generated by our method on RPN-vgg16. The performance of instance segmentation is evaluated using the mAP metric, replacing the IoU of detection boxes with the IoU of masks. We use threshold 0.5 in this experiment. Table 5 shows the performance of these methods on original images and adversarial images. We can observe the performance is notably degraded to almost zero for both instance segmentation methods, which demonstrates the strong transferability of our method on instance segmentation task. Fig. 4 illustrates several examples of our method on attacking Mask-RCNN and YOLACT.

4.7. Real-world API

We further validate our method on a real-world API released by Facebook named `Detectron2`[5], which implements the state-of-the-art object detection and instance segmentation methods. We select the Mask-RCNN option with ResNet50-FPN (Lin et al., 2017a) base network. To evaluate the performance of our method, we use the predicted results of Detectron2 on original images as the ground truth. Since the API only returns the final results after post processing, the mAP metric is not suitable in this case. Therefore, we calculate the accuracy instead. Specifically, the detection is correct if the overlap (IoU) between the detection and corresponding ground truth is greater

[5] `https://github.com/facebookresearch/detectron2`

Fig. 9: Visual examples of our method on Detectron2.

than a threshold 0.5 (same in instance segmentation evaluation). Our method can achieve accuracy 0 in both detection and instance segmentation. On one hand, it demonstrates the strong transferability of our method. On the other hand, it also reveals that the robustness against adversarial attack is not fully considered in Detectron2. Fig. 9 shows several examples of our method on attacking Detectron2.

5. Conclusion

We describe a new method, namely TransRPN, to generate adversarial perturbations transferring among different object detectors – different categories (*e.g.*, SSD, Faster-RCNN, YOLO) as well as different base networks (*e.g.*, VGG16, ResNet, MobileNet), and also in other tasks such as instance segmentation methods. Our method focus on attacking the Region Proposal Network (RPN) by disrupting the intermediate feature. Thanks to the property of RPNs that can take images with arbitrary size, our method can directly attack the original input image without changing the size. The experiments are conducted on MSCOCO dataset with nine object detectors and two instance segmentation methods, as well as one real-world API Detectron2, which demonstrate the strong transferability of our method.

The future work will focus on exploring the transferable adversarial attack on the most recent anchor-free object detectors such as (Zhou et al., 2019; Duan et al., 2019; Tian et al., 2020; Kong et al., 2020). The anchor-free object detectors directly regress the center location of each object, which does not predict the offset of each anchor box used in RPN. Inspired by TransRPN, we would like to investigate the effect of intermediate features to different anchor-free object detectors.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1816227 and IIS-2008532.

References

Akhtar, N., Mian, A., 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. IEEE Access .

Arnab, A., Miksik, O., Torr, P.H., 2018. On the robustness of semantic segmentation models to adversarial attacks, in: IEEE Conference on Computer Vision and Pattern Recognition.

Baluja, S., Fischer, I., 2018. Learning to attack: Adversarial transformation networks, in: Association for the Advancement of Artificial Intelligence.

Bolya, D., Zhou, C., Xiao, F., Lee, Y.J., 2019. Yolact: Real-time instance segmentation, in: IEEE International Conference on Computer Vision.

Brendel, W., Rauber, J., Bethge, M., 2017. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. arXiv preprint arXiv:1712.04248 .

Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., Mukhopadhyay, D., 2018. Adversarial attacks and defences: A survey. arXiv preprint arXiv:1810.00069 .

Chen, S.T., Cornelius, C., Martin, J., Chau, D.H., 2018. Robust physical adversarial attack on faster R-CNN object detector. arXiv preprint arXiv:1804.05810 .

Chow, K.H., Liu, L., Gursoy, M.E., Truex, S., Wei, W., Wu, Y., 2020a. Tog: Targeted adversarial objectness gradient attacks on real-time object detection systems. arXiv preprint arXiv:2004.04320 .

Chow, K.H., Liu, L., Gursoy, M.E., Truex, S., Wei, W., Wu, Y., 2020b. Understanding object detection through an adversarial lens. arXiv preprint arXiv:2007.05828 .

Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J., 2018. Boosting adversarial attacks with momentum, in: IEEE Conference on Computer Vision and Pattern Recognition.

Dong, Y., Pang, T., Su, H., Zhu, J., 2019. Evading defenses to transferable adversarial examples by translation-invariant attacks, in: IEEE Conference on Computer Vision and Pattern Recognition.

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q., 2019. Centernet: Keypoint triplets for object detection, in: IEEE International Conference on Computer Vision.

Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge. International journal of computer vision .

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, in: Conference on Neural Information Processing Systems.

Goodfellow, I.J., Shlens, J., Szegedy, C., 2015. Explaining and harnessing adversarial examples, in: International Conference on Learning Representations.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask R-CNN, in: IEEE International Conference on Computer Vision.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv 1704.04861 .

Huan, Z., Wang, Y., Zhang, X., Shang, L., Fu, C., Zhou, J., 2020. Data-free adversarial perturbations for practical black-box attack, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining.

Huang, Z., Zhang, T., 2020. Black-box adversarial attack with transferable model-based embedding, in: International Conference on Learning Representations.

Kong, T., Sun, F., Liu, H., Jiang, Y., Li, L., Shi, J., 2020. Foveabox: Beyound anchor-based object detection. IEEE Transactions on Image Processing .

Kurakin, A., Goodfellow, I., Bengio, S., 2017. Adversarial examples in the physical world, in: International Conference on Learning Representations.

Li, D., Zhang, J., Huang, K., 2020a. Universal adversarial perturbations against object detection. Pattern Recognition .

Li, Q., Guo, Y., Chen, H., 2020b. Yet another intermediate-level attack, in: European Conference on Computer Vision.

Li, Y., Bai, S., Zhou, Y., Xie, C., Zhang, Z., Yuille, A., 2020c. Learning transferable adversarial examples via ghost networks, in: Association for the Advancement of Artificial Intelligence.

Li, Y., Bian, X., Chang, M., Lyu, S., 2019. Exploring the vulnerability of single shot module in object detectors via imperceptible background patches, in: BMVC.

Li, Y., Tian, D., Chang, M., Bian, X., Lyu, S., 2018. Robust adversarial perturbation on deep proposal-based models, in: BMVC.

Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., Zhu, J., 2018. Defense against adversarial attacks using high-level representation guided denoiser, in: IEEE Conference on Computer Vision and Pattern Recognition.

Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017a. Feature pyramid networks for object detection, in: IEEE Conference on Computer Vision and Pattern Recognition.

Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017b. Focal loss for dense object detection, in: IEEE International Conference on Computer Vision.

Lin, T.Y., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft COCO: Common objects in context, in: European Conference on Computer Vision.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C., 2016. SSD: Single shot multibox detector, in: European Conference on Computer Vision.

Liu, Y., Chen, X., Liu, C., Song, D., 2017. Delving into transferable adversarial examples and black-box attacks, in: International Conference on Learning Representations.

Lu, Y., Jia, Y., Wang, J., Li, B., Chai, W., Carin, L., Velipasalar, S., 2020. Enhancing cross-task black-box transferability of adversarial examples with dispersion reduction, in: IEEE Conference on Computer Vision and Pattern Recognition.

Luo, B., Liu, Y., Wei, L., Xu, Q., 2018. Towards imperceptible and robust adversarial example attacks against neural networks, in: Association for the

Advancement of Artificial Intelligence.

Luo, Y., Boix, X., Roig, G., Poggio, T., Zhao, Q., 2015. Foveation-based mechanisms alleviate adversarial examples. arXiv preprint arXiv:1511.06292 .

Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P., 2017. Universal adversarial perturbations, in: IEEE Conference on Computer Vision and Pattern Recognition.

Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P., 2016. Deepfool: a simple and accurate method to fool deep neural networks, in: IEEE Conference on Computer Vision and Pattern Recognition.

Mopuri, K.R., Ganeshan, A., Babu, R.V., 2018. Generalizable data-free objective for crafting universal adversarial perturbations. IEEE Transactions on Pattern Analysis and Machine Intelligence .

Mustafa, A., Khan, S.H., Hayat, M., Goecke, R., Shen, J., Shao, L., 2020. Deeply supervised discriminative learning for adversarial defense. IEEE Transactions on Pattern Analysis and Machine Intelligence .

Naseer, M., Khan, S., Hayat, M., Khan, F.S., Porikli, F., 2020. A self-supervised approach for adversarial robustness, in: IEEE Conference on Computer Vision and Pattern Recognition.

Ozbulak, U., Vandersmissen, B., Jalalvand, A., Couckuyt, I., Van Messem, A., De Neve, W., 2021. Investigating the significance of adversarial attacks and their relation to interpretability for radar-based human activity recognition systems. Computer Vision and Image Understanding .

Pang, T., Du, C., Dong, Y., Zhu, J., 2018. Towards robust detection of adversarial examples, in: Conference on Neural Information Processing Systems.

Papernot, N., McDaniel, P., Goodfellow, I., 2016a. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. arXiv preprint arXiv:1605.07277 .

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A., 2016b. The limitations of deep learning in adversarial settings, in: EuroS&P.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., 2019. Pytorch: An imperative style, high-performance deep learning library, in: Conference on Neural Information Processing Systems.

Poursaeed, O., Katsman, I., Gao, B., Belongie, S., 2018. Generative adversarial perturbations, in: IEEE Conference on Computer Vision and Pattern Recognition.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection, in: IEEE Conference on Computer Vision and Pattern Recognition.

Redmon, J., Farhadi, A., 2017. Yolo9000: better, faster, stronger, in: IEEE Conference on Computer Vision and Pattern Recognition.

Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 .

Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence .

Samangouei, P., Kabkab, M., Chellappa, R., 2018. Defense-gan: Protecting classifiers against adversarial attacks using generative models, in: International Conference on Learning Representations.

Serban, A., Poll, E., Visser, J., 2020. Adversarial examples on object recognition: A comprehensive survey. ACM Computing Surveys (CSUR) .

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 .

Stutz, D., Hein, M., Schiele, B., 2020. Confidence-calibrated adversarial training: Generalizing to unseen attacks, in: International Conference on Machine Learning.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R., 2014. Intriguing properties of neural networks, in: International Conference on Learning Representations.

Tian, Z., Shen, C., Chen, H., He, T., 2020. Fcos: A simple and strong anchor-free object detector. IEEE Transactions on Pattern Analysis and Machine Intelligence .

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P., 2017. Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204 .

Wang, Y., Wang, K., Zhu, Z., Wang, F.Y., 2020. Adversarial attacks on faster r-cnn object detector. Neurocomputing .

Wei, X., Liang, S., Chen, N., Cao, X., 2019. Transferable adversarial attacks for image and video object detection, in: International Joint Conferences on Artificial Intelligence.

Wu, D., Wang, Y., Xia, S.T., Bailey, J., Ma, X., 2020a. Skip connections matter: On the transferability of adversarial examples generated with resnets, in: International Conference on Learning Representations.

Wu, W., Su, Y., Chen, X., Zhao, S., King, I., Lyu, M.R., Tai, Y.W., 2020b. Boosting the transferability of adversarial samples via attention, in: IEEE Conference on Computer Vision and Pattern Recognition.

Wu, Z., Lim, S.N., Davis, L.S., Goldstein, T., 2020c. Making an invisibility cloak: Real world adversarial attacks on object detectors, in: European Conference on Computer Vision.

Xie, C., Wang, J., Zhang, Z., Ren, Z., Yuille, A., 2018. Mitigating adversarial effects through randomization, in: International Conference on Learning Representations.

Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., Yuille, A., 2017. Adversarial examples for semantic segmentation and object detection, in: IEEE International Conference on Computer Vision.

Xie, C., Zhang, Z., Zhou, Y., Bai, S., Wang, J., Ren, Z., Yuille, A.L., 2019. Improving transferability of adversarial examples with input diversity, in: IEEE Conference on Computer Vision and Pattern Recognition.

Zeng, X., Liu, C., Qiu, W., Xie, L., Tai, Y.W., Tang, C.K., Yuille, A.L., 2017. Adversarial attacks beyond the image space. arXiv 1711.07183 .

Zhang, B., Tondi, B., Barni, M., 2020a. Adversarial examples for replay attacks against cnn-based face recognition with anti-spoofing capability. Computer Vision and Image Understanding .

Zhang, H., Zhou, W., Li, H., 2020b. Contextual adversarial attacks for object detection, in: ICME.

Zhou, M., Wu, J., Liu, Y., Liu, S., Zhu, C., 2020. Dast: Data-free substitute

training for adversarial attacks, in: IEEE Conference on Computer Vision and Pattern Recognition.

Zhou, W., Hou, X., Chen, Y., Tang, M., Huang, X., Gan, X., Yang, Y., 2018. Transferable adversarial perturbations, in: European Conference on Computer Vision.

Zhou, X., Wang, D., Krähenbühl, P., 2019. Objects as points. arXiv preprint arXiv:1904.07850 .