G-UAP: Generic Universal Adversarial Perturbation that Fools RPN-based Detectors

Xing Wu WUXING6@MAIL2.SYSU.EDU.CN
Lifeng Huang HUANGLF6@MAIL2.SYSU.EDU.CN
Chengying Gao* MCSGCY@MAIL.SYSU.EDU.CN
School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China

Editors: Wee Sun Lee and Taiji Suzuki

Abstract

Adversarial perturbation constructions have been demonstrated for object detection, but these are image-specific perturbations. Recent works have shown the existence of image-agnostic perturbations called universal adversarial perturbation (UAP) that can fool the classifiers over a set of natural images. In this paper, we extend this kind perturbation to attack deep proposal-based object detectors. We present a novel and effective approach called G-UAP to craft universal adversarial perturbations, which can explicitly degrade the detection accuracy of a detector on a wide range of image samples. Our method directly misleads the Region Proposal Network (RPN) of the detectors into mistaking foreground (objects) for background without specifying an adversarial label for each target (RPN's proposal), and even without considering that how many objects and object-like targets are in the image. The experimental results over three state-of-the-art detectors and two datasets demonstrate the effectiveness of the proposed method and transferability of the universal perturbations.

Keywords: Universal adversarial perturbation (UAP), image-agnostic, RPN-based detectors, transferability

1. Introduction

Deep Neural Networks (DNN) achieve superior performance in many problems in computer vision, including image classification, object detection and semantic segmentation, etc. Although deep networks provide the state-of-the-art performance in many tasks, it has shown that samples which are maliciously altered affect the networks' prediction drastically. Szegedy et al. (2014) first showed that adding visually imperceptible perturbations to inputs can result in failures for image classification. To date, there has been large effort in investigating the existence of adversarial perturbations (Goodfellow et al. (2015); Nguyen et al. (2015); Moosavi-Dezfooli et al. (2017)). It has been shown that many DNN-based algorithms are vulnerable to adversarial perturbations, which can fool the system into inferring wrong predictions, but they are imperceptible to humans. Investigating adversarial perturbations not only contributes to understanding the working mechanism of DNN, but also offers opportunities to improve the robustness of networks.

* Corresponding author

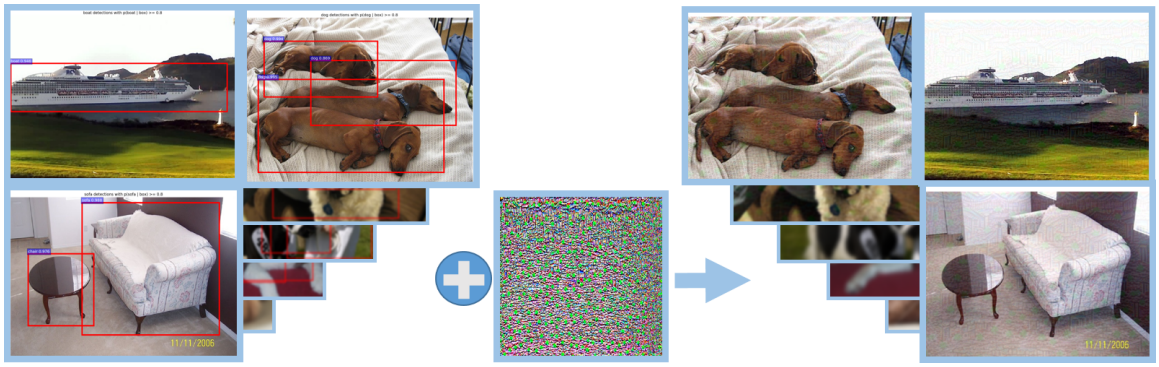


Figure 1: Left images: detection results of original natural images. Central image: universal adversarial perturbation crafted by our method. Right images: detection results of perturbed images. When the perturbation is added to a natural image, the detector can not detect the objects.

Multiple researches (Goodfellow et al. (2015); Kurakin et al. (2017); Moosavi-Dezfooli et al. (2016)) have focused on fooling image classifiers. Goodfellow et al. (2015) proposed the fast gradient sign method (FGSM) to generate adversarial perturbations based on the linear nature of DNN. Moosavi-Dezfooli et al. (2016) presented an algorithm to compute the minimal adversarial perturbation. Recently, developing approaches to attack deep neural networks beyond classification has attracted many attentions. Xie et al. (2017) proposed Dense Adversary Generation (DAG) to compute adversarial perturbations for semantic segmentation and object detection. Lu et al. (2017) attempted to generate adversarial perturbations on traffic sign and human face to mislead detectors. Li et al. (2018) proposed the robust adversarial perturbation (R-AP) method which focuses on attacking RPN of deep proposal-based models without knowing the details of models' architecture. However, these methods are image-specific. It means that causing a specific input to be incorrect prediction they need to regenerate specific perturbation. This kind perturbation has two limitations: poor transferability, which implies that the specific adversarial perturbation cannot attack other images, and high computation cost, which means that it takes time to generate an adversarial perturbation for each image respectively.

Different from the above methods, Moosavi-Dezfooli et al. (2017) showed the existence of a single perturbation that when added to most of images, could cause wrong predictions. It is referred to as universal adversarial perturbation. Similar to Moosavi-Dezfooli et al. (2017), Metzen et al. (2017) developed UAP for semantic segmentation task. Mopuri *et al.* showed their techniques (FFF Mopuri et al. (2017), GDUAP Mopuri et al. (2018)) to craft universal adversarial perturbations that can fool the target model without prior knowledge about the dataset. They demonstrated that their crafted perturbations were transferable to three different vision tasks covering classification, depth estimation and segmentation.

In this paper, we go one step further to propose a generic universal adversarial perturbation (G-UAP) method to craft universal adversarial perturbations to fool detectors. Specifically, by adding such an unique perturbation to natural images, the prediction esti-

mated by RPN-based detectors is wrong (see Fig. 1). The proposals predict that the objects in images are all backgrounds, so there are no bounding boxes marked in the images. Similar to Xie et al. (2017); Li et al. (2018), our G-UAP algorithm also focuses on attacking detection models based on RPN and we mainly study Faster-RCNN Ren et al. (2017) which is one of the state-of-the-art object detectors. Xie et al. (2017) assigned an adversarial label for each proposal region and then performed iterative gradient back-propagation to misclassify the proposals. The similar method is also presented in Li et al. (2018). Besides, Wei et al. (2019) proposed the Unified and Efficient Adversary (UEA) for image and video object detection. For universal adversarial prediction and many unknown images, it is impractical to assign adversarial labels to proposals. Because an image has many objects and object-like targets (proposals) and for detectors based on RPN, one object in an image has multiple targets to predict it. Although we successfully fool a target, any other targets can still correctly predict this object. Considering that there are orders of magnitude more targets in an image, we propose to mislead RPN into wrongly classifying a target uniformly. In other words, attacking RPN to make targets misclassify the foreground as the background as much as possible no matter how large the number of targets is. Our experimental results demonstrate that the learned perturbations by G-UAP can significantly degrade the performance of detectors, albeit being imperceptible to human observers. Furthermore, we use the universal adversarial perturbation computed by one network to attack another network in order to investigate the transferability of the generated perturbations. Specially, the Region Fully Convolutional Network (R-FCN) Dai et al. (2016) with RPN network works as a black-box detector to demonstrate cross model generalizability. Besides, Single Shot MultiBox Detector (SSD) Liu et al. (2016) without RPN also works as a black-box detector to test the feasibility of our method in comparison. In addition, an extended experiment has been done. Note that although the pertubations by our method can reduce the detection performance of most images in test set, there are still some images that are hard to fool, especially large objects like people in the images as Fig. 5 shows. As for the failed images, we can use G-UAP to fine-tune the learned universal perturbation exclusively for this image, which is denoted as FG-UAP. And it's noted that this process speeds up the generation of the single specific perturbation.

Our contributions are summarized in the following:

- We propose a novel algorithm called G-UAP that can explicitly degrade performance of RPN by merely disturbing the proposal label prediction of foreground in RPN.
- To the best of our knowledge, G-UAP is the first work to craft universal adversarial perturbations to fool the RPN-based detectors.
- Extensive experiments demonstrate that the proposed algorithm can generate perturbations that exhibit cross model generalizability.

2. Proposed Approach

In this section, we introduce G-UAP algorithm to compute universal adversarial perturbations that can effectively disturb the predictions of RPN-based detectors. The objective is to mislead RPN into wrong predictions about foreground. In others words, making the detectors detect nothing as much as possible. Section 2.1 reviews the conception of RPN. Section 2.2 describes the notations and details for our G-UAP algorithm.

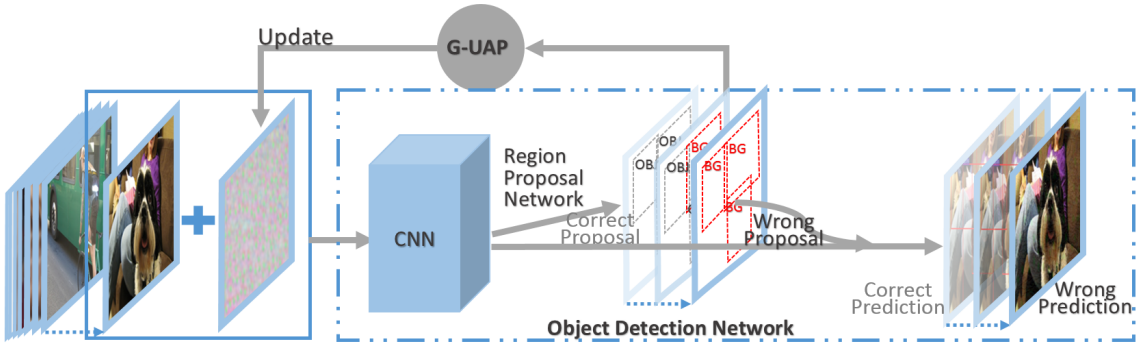


Figure 2: Overview of our G-UAP method. Object Detection Network is easily fooled by perturbed images that are produced by our algorithm which attacks RPN of detectors in order to make it mistake the foreground for the background. We updates the unique perturbation by our loss function mentioned in Sec. 2.2, with iterating through a series of images.

2.1. Region Proposal Network

Region Proposal Network is a kind of fully-convolutional network for object proposal generation. The RPN takes an image as input and outputs confidence scores for object-like targets. The RPN simultaneously predicts k region targets (k is 9 in Faster-RCNN). The layer of binary classification outputs $2k$ scores that predict probability of forground / background for each target.

At training phase, each target is matched to ground-truth label. For RPN, the prediction of target is a problem of binary classfication. Classification loss function for an image is defined as:

$$L(p_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*), \tag{1}$$

where the cls layer is a two-class softmax layer, p_i is the predicted probability of proposal i being an object and p_i^* is the ground-truth label.

At testing phase, the predictions of all targets are generated by cls layer within a single forward. We can get the output of the cls layer when adding perturbation to an image and feeding it to the detector.

2.2. G-UAP Algorithm

First, We formalize the notion followed throughout the paper. Let $\mathcal{X}$ denotes a distribution of images in $\mathbb{R}^d$, f denotes the network of the detector and l_{cls} denotes the two-class softmax layer that maps an input image $x \sim \mathcal{X}$ to its probability output. δ denotes the image-agnostic perturbation learned by our algorithm. Similar to input x, δ also belongs to $\mathbb{R}^d$.

Algorithm. Let $\{x_1, ..., x_m\}$ be a set of images sampled from the distribution $\mathcal{X}$. To craft the adversarial perturbation which is quasi-imperceptible for humans, the pixel intensities of perturbation δ should be restricted. Existing works (eg: Moosavi-Dezfooli

et al. (2017); Mopuri et al. (2017); Mopuri et al. (2018)) imposed an l_∞ constraint ξ ($\xi = 10$ less than 8% of the data range) to control the magnitude of the perturbation to realize imperceptibility.

Our proposed algorithm seeks a universal perturbation δ, such that $\|\delta\|_\infty < \xi$, while leading to wrong predictons at cls layer which predicts foreground or background and thereby misleading the results of the following layers. That is, making RPN mistake the foreground for the background. This problem can be considered as the following optimization problem.

$$\begin{aligned} l_{cls\_obj}(x_i + \delta) &\approx l_{cls\_obj}(x_i) + \mathcal{J}_{l_{cls\_obj}}(x_i)\delta, \\ l_{cls\_bg}(x_i + \delta) &\approx l_{cls\_bg}(x_i) + \mathcal{J}_{l_{cls\_bg}}(x_i)\delta, \end{aligned} \tag{2}$$

where $\mathcal{J}_l(x_i)$ is Jacobian matrix, $l_{cls\_obj}(x_i + \delta)$ is the probability scores of foreground that we want them to be 0 and $l_{cls\_bg}(x_i + \delta)$ is the probability scores of background that we want them to be 1 in the output at layer of cls when $x_i + \delta$ is fed to the network f. Eq. 2 means that adjusting δ to make the right side of the first line equation close to 0 and the second line equation close to 1.

The problem of this perturbation generation can be casted as an optimization problem of binary classification. This binary classification task has only two categories including foreground and background and the sum of the probability is 1, so we only need to reduce the probability of the foreground. For this classification problem, we use Cross Entropy Loss as our loss function:

$$L(\delta) = -[l \log(\hat{l}(x_i + \delta)) + (1 - l) \log(1 - \hat{l}(x_i + \delta))], \tag{3}$$

where l represents the label 0 specified for the foreground and $\hat{l}(x_i+\delta)$ is same as $l_{cls\_obj}(x_i+\delta)$ metioned above. Thus, when l is equal to 0, the Eq. 3 becomes:

$$L(\delta) = -\log(1 - \hat{l}(x_i + \delta)). \tag{4}$$

Minimizing this loss is equivalent to decreasing confidence score of foreground and increasing confidence score of background in cls layer. Then the objective is to solve for:

$$\delta^* = \arg\min_{\delta} L(\delta). \tag{5}$$

Algorithm 1 presents the detailed algorithm. We apply a gradient descent algorithm for optimization. The re-scaling optimization by Mopuri et al. (2018) avoid accumulating δ beyond the imposed max-norm constraint (ξ). In particular, we randomly choose 1000 images from PascalVOC-2012 training set as data points x. And we will examine the influence of the size of $\mathcal{X}$ on the quality of the universal perturbations in Sec. 3.4.

Algorithm 1: Computation of universal perturbations.
Input: Data points $\mathcal{X}$, Network f
Output: Universal perturbation vector δ, Mean average precision mAP
Initialize $\delta \leftarrow$ U(-10,10).
Initialize mAP $\leftarrow$ 100.
Initialize Stop $\leftarrow$ 0.
while *Stop<10* **do**
  for *each datapoint* $x_i \in \mathcal{X}$ **do**
    $\Delta\delta_i \leftarrow \nabla L(\delta)$
    Perform adaptive re-scaling on δ_i
    Compute current_mAP on dataset V
    if *current_mAP<mAP* **then**
      mAP $\leftarrow$ current_mAP
      $\delta \leftarrow \delta_i$
    end
    else
      Stop$\leftarrow$Stop+1
    end
  end
end
return δ, mAP

3. Experiments

In this section, we evaluate our approach on several state-of-the-art object detectors. Section 3.1 describes implementation settings of our G-UAP algorithm. Section 3.2 compares our results with the baseline models on object detectors. Section 3.3 investigates the transferability of the generated perturbations. Section 3.4 analyses the performance of our algorithm.

3.1. G-UAP Setup

We evaluate our method by measuring the drop in detection accuracy using the original test images and the ones after adding adversarial perturbations. Mean average precision (mAP) is an evaluation criterion for object detection and Peak Signal-to-Noise Ratio (PSNR) is used as an approximation of image quality. Less perturbation results in higher PSNR. We study four state-of-the-art object detectors, including two Faster-RCNN models based on the 16-layer VGGNet Simonyan and Zisserman (2015), 101-layer ResNet He et al. (2016), the Region Fully Convolutional Network (R-FCN) Dai et al. (2016) based on the 101-layer ResNet and VGG-based Single Shot MultiBox Detector (SSD) Liu et al. (2016). The Faster-RCNN object detectors we use are implemented by Chen and Gupta (2017). Faster-RCNN are either trained on the PascalVOC-2007 trainval set or PascalVOC-0712 trainval set. These four models are denoted as FR-V16-07, FR-V16-0712, FR-R101-07 and FR-R101-0712, respectively. We use the PascalVOC-2007 test set (V in Algorithm 1) which has 4952 images to evaluate our algorithm. Besides, we also select a KITTI dataset with 156 consecutive frames to learn the perturbation and verify the universality of our approach.

Similar to the existing approaches (Moosavi-Dezfooli et al. (2017); Mopuri et al. (2017, 2018)), we restrict the pixel intensities of the perturbation to lie within [-10,+10] range by choosing ξ metioned in Sec. 2.2 to be 10.

(a)GNRM[-10,10] (b)GNRM[-65,65] (c)GDUAP[-10,10] (d)G-UAP[-10,10]

Figure 3: Four kind perturbed images. (a)(b) by adding Gaussian noise perturbations with different pixel intensities, (c)(d) by GD-UAP and our G-UAP algorithm from FR-V16-07. PSNR is displayed in each image.

Table 1: Performance of four kind perturbations on detection accuracy degradation (measured by mAP, %). ORIG represents the accuracy obtained on the original image set D. GNRMs are obtained after Gaussian noises with different pixel intensities are added. GDUAP is obtained by baseline model and G-UAP represents the accuracy obtained by our method.

Network	ORIG	GNRM-10	GNRM-65	GDUAP	G-UAP
FR-V16-07	70.8	69.1	30.4	47.7	**31.2**
FR-V16-0712	75.7	73.9	41.4	57.3	**33.7**

3.2. Experimental Results

In order to bring out the effectiveness of our image-agnostic perturbations, we compare the performance of our learned perturbations with the state-of-the-art method GD-UAP Mopuri et al. (2018). The reason why we select this baseline is because there is no method of craft universal perturbation especially for fooling detectors to compare. And GD-UAP is generalizable across different vision tasks, so we apply it to fooling detectors and observe its performance. In addition, we also generate Gaussian noise (random) perturbations as baselines to demonstrate the effectiveness of G-UAP in comparison.

Fig. 3 shows Gaussian noise (random) perturbations, image-agnostic perturbations δ crafted by GDUAP and our proposed method. Table 1 summarizes degradation of the detection performance after applying four perturbations to images. The GNRM-10 column of the table shows that adding random Gaussian noise with same pixel intensity as the perturbation to be ineffective in attacking, with only $< 2\%$ performance degradation. Although GNRM-65 can degrade the performance of detetors closing to ours as Table 1 shows, its PSNR is too low. In contrast, our G-UAP leads to large degradation of the detection

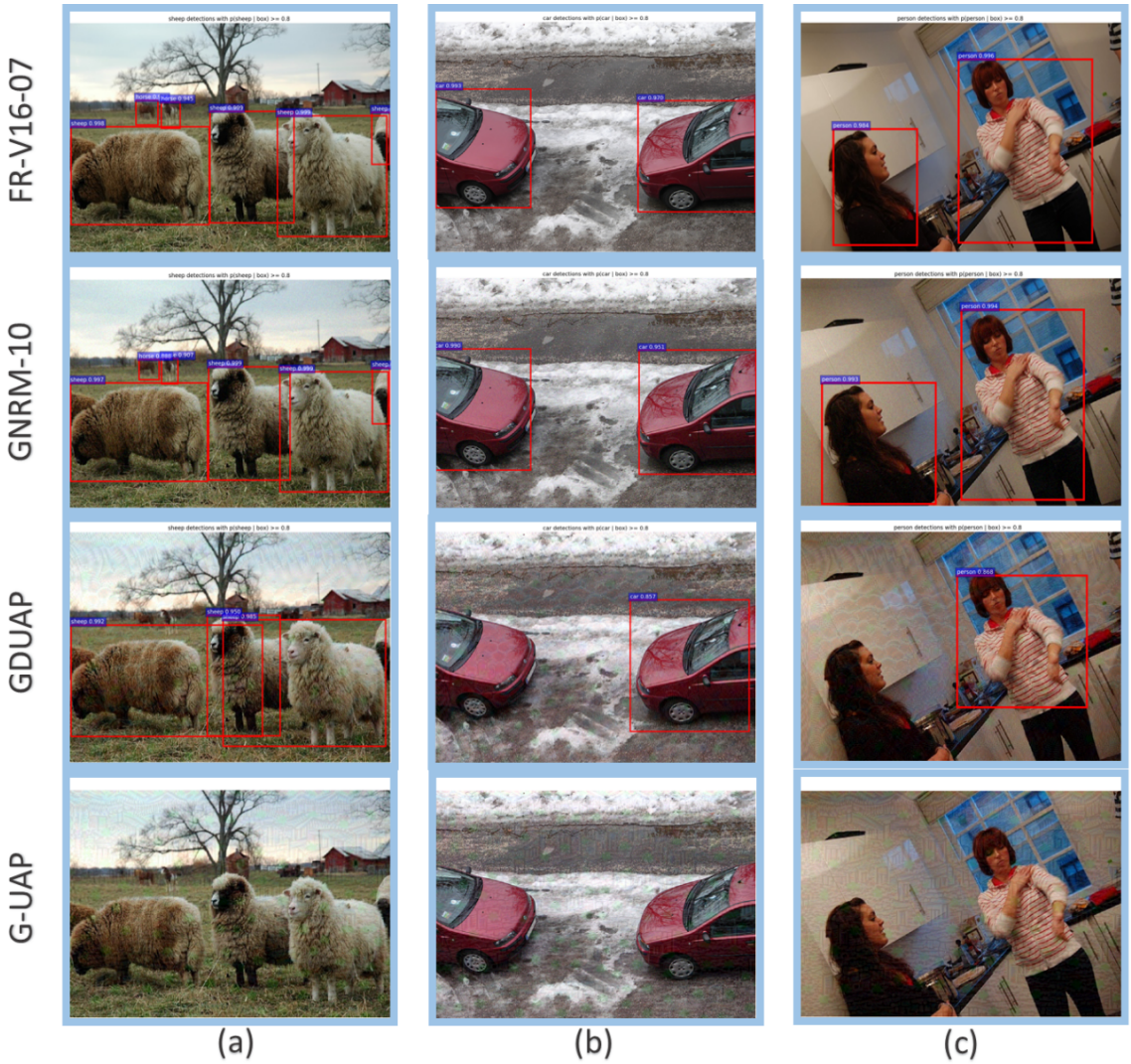


Figure 4: Detection results of sample perturbed images by various methods. The first row shows the detection results of the original images. And the rest rows show the detection results of corresponding adversarial images.

performance than GDUAP in both FR-V16-07 and FR-V16-0712. And note that the perturbation crafted by GDUAP looks more conspicuous than perturbation by G-UAP (the latter's PSNR is higher).

In Fig. 4, all the clean images shown in the figure are correctly detected. Note that adversarial images can still be correctly detected when adding Gaussian noise (random) perturbation to clean images. There are still partially correct detections when adding perturbation by GDUAP. In contrast, perturbations crafted by our G-UAP algorithm can lead the detector to detecting nothing for most images. As for the failed images, we can use G-UAP to fine-tune the learned universal perturbation exclusively for this image, which is denoted as FG-UAP. Fig. 5 shows that the failed targets can get fooled entirely after

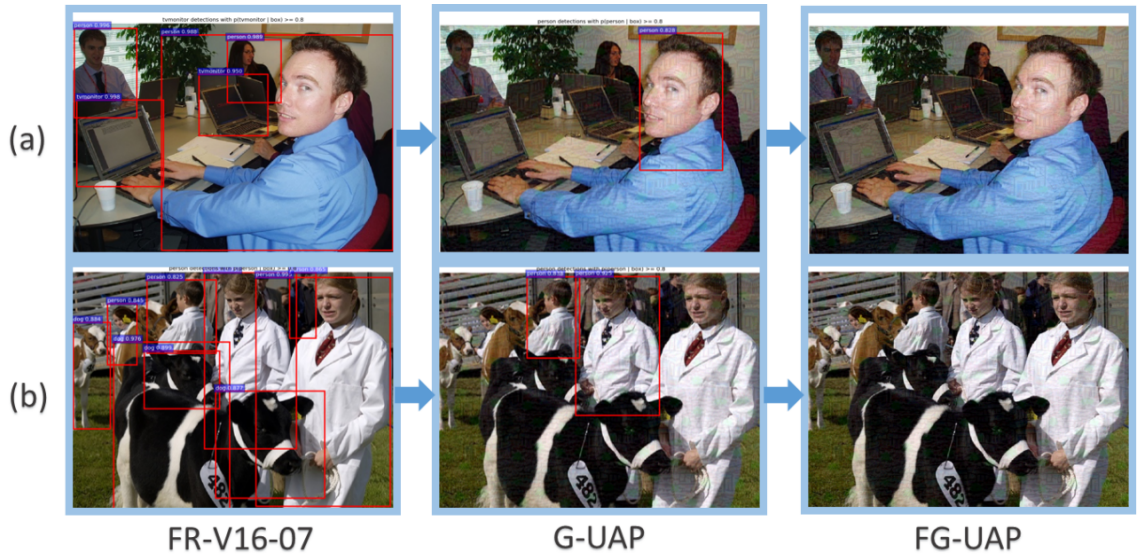


Figure 5: The fine-tuning results of our method for the failed images.

we fine-tune the perturbation and it costs little time (5∼10 iterations less than 150∼200 iterations for taking a start from the head to compute a specific perturbation by DAG Xie et al. (2017)). Experiments show that by fine-tuning all failed targets, we can get nearly 0% mAP. This demonstrates that the universal perturbation can be used as an initialization to speed up the generation of the single specific perturbation. Fig. 6 presents the detection results of another dataset. We select one image every ten frames to examine the detection performance on adversarial examples in order to demonstrate the effectiveness of our algorithm for attacking tracking detection. These results demonstrate that perturbations crafted by our method achieve excellent attacking performance on different datasets.

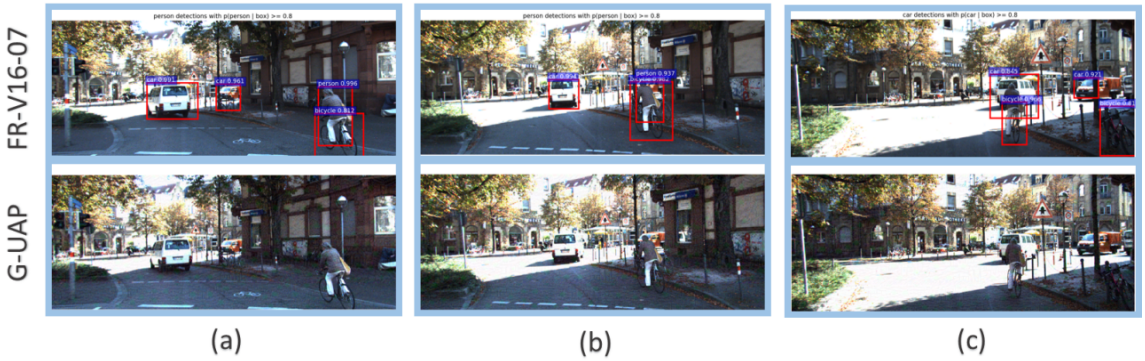
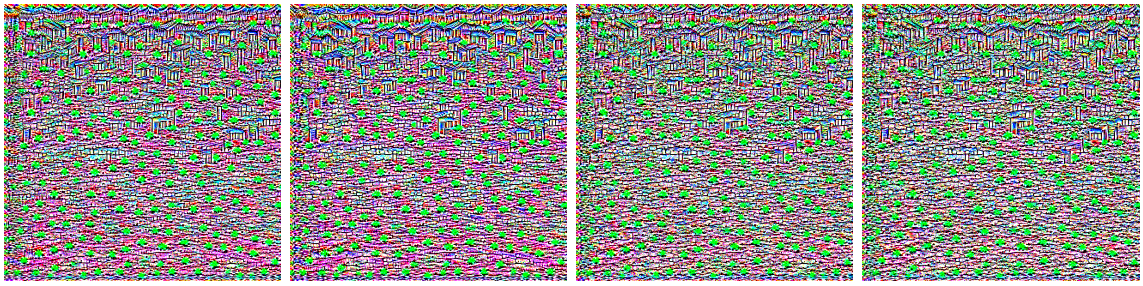


Figure 6: First row presents detection results of clean images from KITTI dataset. Second row shows the results of adversarial examples. In particular, the perturbation is learned from KITTI.



(a)FR-V16-07 (b)FR-V16-0712 (c)FR-R101-07 (d)FR-R101-0712

Figure 7: Universal adversarial perturbations crafted by our G-UAP algorithm. Corresponding target network architecture is mentioned below each image.

Table 2: Performance of G-UAP on 6 state-of-the-art object detectors. ORIG column represents the accuracy of five models obtained on the original dataset D. Besides, each row of the table shows the detection accuracies for adding the perturbations learned from the Faster-RCNN models (column).

Network	ORIG	V16-07	V16-0712	R101-07	R101-0712
V16-07	70.8	**31.2**	28.3	34.7	35.1
V16-0712	75.7	34.1	**33.7**	44.3	44.2
R101-07	75.7	57.0	56.0	**50.3**	46.9
R101-0712	79.8	63.4	62.3	56.9	**52.8**
R-FCN	73.8	50.4	49.4	48.2	45.4
SSD	77.8	70.1	67.7	71.0	71.2

3.3. Transferability Across Network

We now examine perturbations' cross-model universality. We compute universal perturbation on one specific network and observe its ability to fool other networks. Fig. 7 shows the universal perturbations δ obtained for the networks by using our proposed method. Note that the perturbations are visually different for each network architecture. Table 2 illustrates the performance of perturbations generated from four different object models by G-UAP. Each column in the Table 2 indicates one target model employed to learn perturbations and the rows indicate that various models are attacked using the learned perturbations except for ORIG column. Diagonal values in bold are white-box attack. In addition, the R-FCN and SSD work as black-box detectors in our experiment. Results show that the learned algorithm can generate perturbations that exhibit cross model generalizability. The perturbations generated from one model can effectively reduce the detection performance of other models and perturbations crafted by robust model have stronger attack ability. It can be observed that universal perturbations computed for FR-ResNet101 are more valid for FR-VGG16. Besides, observing that in the R-FCN row of Table 2, the perturbations generated from V16-07, V16-0712, R101-07 and R101-0712 of Faster-RCNN can effectively reduce the detection performance of R-FCN. But for SSD, there is no significant effect. The

main reason is related to the network structure of the SSD, which is not based on RPN network. However, our algorithm mainly aims at attacking RPN.

3.4. Discussion and Analysis

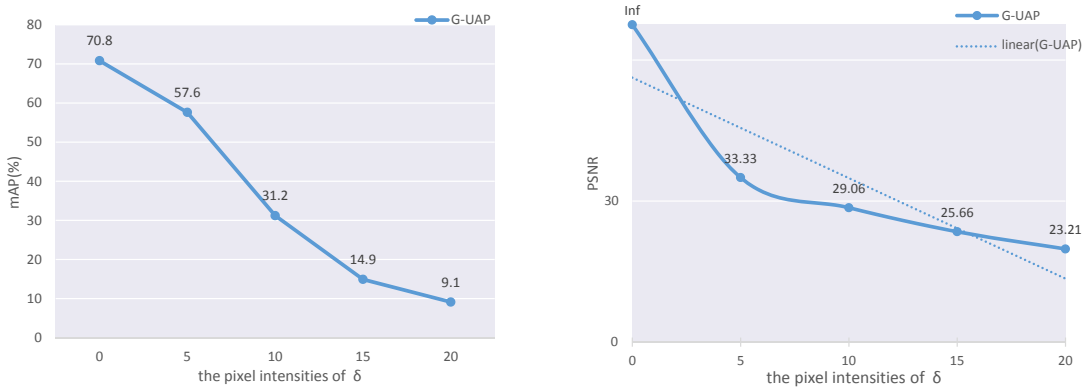
(a) (b)

Figure 8: (a)quantities of object prediction for original images and adversarial images. (b)the influence of $\mathcal{X}$'s size on mAP.

In this section, to gain insights on the effect of universal perturbations on images, we visualize our method to prove the validity of the algorithm. First, we randomly choose 100 images to observe the change of quantity of object prediction before and after adding the perturbation. Second, we examine the influence of the size of $\mathcal{X}$ on the decrease of detection accuracy. Besides, we adjust pixel intensities of δ to see the effect it has on mAP. Finally, we analyse the effect of class label distribution of the selected training data. Here the notion of 'Fooling rate' has been well defined for the degradation percentage of detection accuracy for each class.

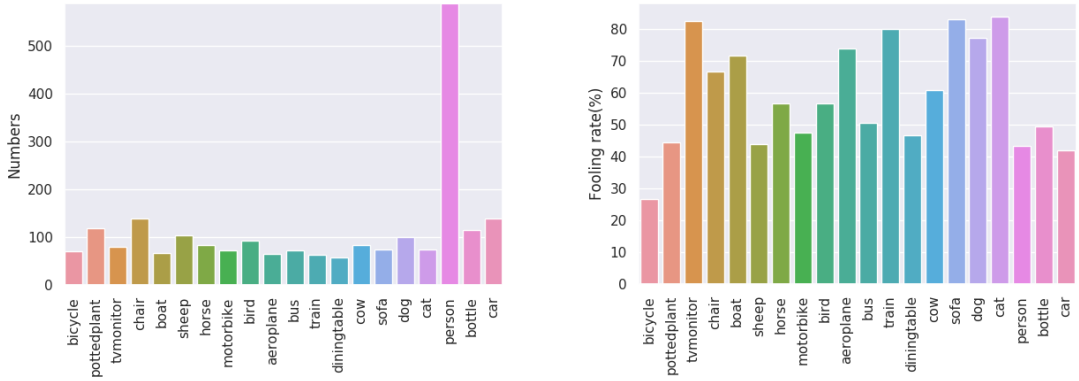
Fig. 8 (a) shows that the number of object prediction significantly decreases and the detector detects nothing in most images after adding the perturbation. Most of the points in G-UAP are distributed on the 0-scale line, which means that foreground predictions of proposals are misled into background largely. We attribute this ability of our algorithm to the effectiveness of the proposed objective in the loss. Fig. 8 (b) demonstrates that no matter how large the size of $\mathcal{X}$ is, the mAP for baseline model is still high. This figure also proves that our method only need a small number of data points to achieve good performance.

Fig. 9 (a) shows that when the intensity of pertubation δ increases, mAP decreases. But (b) suggest that it is not possible to increase the pertubation's intensity blindly, which will deteriorate the image quality (The dotted line indicates that the PSNR is decreasing when δ is increasing.). Similar to works (Moosavi-Dezfooli et al. (2017); Mopuri et al. (2017); Mopuri et al. (2018)), we impose an l_∞ constraint ξ ($\xi = 10$ less than 8% of the data range) to control the magnitude of the perturbation to ensure PSNR at a higher value in order to make perturbation quasi-imperceptible for humans.



(a) (b)

Figure 9: (a)llustration of mAP under different δ value. (b)llustration of PSNR under different δ value.



(a) (b)

Figure 10: (a)class label distribution of the selected training data $\mathcal{X}$. (b)'Fooling rate (%)' for different classes

Comparing the Fig 10 (a) and (b), we find that though the number of person class is highest, its 'Fooling rate' is not the highest one. The same is true for many other classes. The main reason is that the feature complexity of each class is different. For example, the characteristics of sheep are simple and the characteristics of people are complex. Simple objects are easy to fool, but complex objects are different. The universal perturbation cannot take into account every feature of person class.

4. CONCLUSION

In this paper, we show the existence of small universal perturbations that can fool state-of-the-art detectors on natural images. To the best of our knowledge, this work is the first to investigate the universal adversarial perturbations for attacking Object Detection Networks. Our G-UAP algorithm focuses on attacking RPN of detectors to mislead RPN into inferring

wrong prediction for foreground. Experiments demonstrate that our algorithm significantly reduce detection performance of state-of-the-art object detectors. Furthermore, it has been shown that the learned universal perturbation can be used as an initialization to speed up the generation of the single specific perturbation. This work may reveal something about what is important to a detector and opens up a new opportunity on how to effectively improve the robustness of RPN-based detectors.

However, our method has some limits. There is still much room for improvement in the effects of universal adversarial perturbations. Besides, our generated adversarial pertubations have low success rate to attack other detection methods without RPN. Therefore, it is an important research direction to be focused on building generic methods to attack all kinds of detetors in the future.

Acknowledgments

We thank the reviewers for their valuable comments. This work was supported by the Fundamental Research Funds for the Central Universities, Guangzhou Science Technology and Innovation Commission (GZSTI16EG14/201704030079).

References

Xinlei Chen and Abhinav Gupta. An implementation of faster rcnn with study for region sampling. *arXiv preprint arXiv:1702.02138*, 2017.

Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, 2016.

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

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

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *ICLR*, 2017.

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

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, and Scott Reed. Ssd: Single shot multibox detector. *ECCV*, 2016.

Jiajun Lu, Hussein Sibai, and Evan Fabry. Adversarial examples that fool detectors. *arXiv preprint arXiv:1712.02494*, 2017.

Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, and Volker Fischer. Universal adversarial perturbations against semantic image segmentation. In *ICCV*, 2017.

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

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

Konda Reddy Mopuri, Utsav Garg, and R. Venkatesh Babu. Fast feature fool: A data independent approach to universal adversarial perturbations. In *BMVC*, 2017.

Konda Reddy Mopuri, Aditya Ganeshan, and R. Venkatesh Babu. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2018.

Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 2015.

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

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

Xingxing Wei, Siyuan Liang, Ning Chen, and Xiaochun Cao. Transferable adversarial attacks for image and video object detection. 2019.

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