Transferable Adversarial Attacks for Image and Video Object Detection

Xingxing Wei[1†*], **Siyuan Liang**[2†], **Ning Chen**[1*] and **Xiaochun Cao**[2]

[1]Dept. of Comp. Sci. & Tech., Institute for Artificial Intelligence,
State Key Lab for Intell. Tech. & Sys., THBI Lab, Tsinghua University
[2]Institute of Information Engineering, Chinese Academy of Sciences
{xwei11, ningchen}@mail.tsinghua.edu.cn, {liangsiyuan, caoxiaochun}@iie.ac.cn

Abstract

Identifying adversarial examples is beneficial for understanding deep networks and developing robust models. However, existing attacking methods for image object detection have two limitations: *weak transferability*—the generated adversarial examples often have a low success rate to attack other kinds of detection methods, and *high computation cost*—they need much time to deal with video data, where many frames need polluting. To address these issues, we present a generative method to obtain adversarial images and videos, thereby significantly reducing the processing time. To enhance transferability, we manipulate the feature maps extracted by a feature network, which usually constitutes the basis of object detectors. Our method is based on the Generative Adversarial Network (GAN) framework, where we combine a high-level class loss and a low-level feature loss to jointly train the adversarial example generator. Experimental results on PASCAL VOC and ImageNet VID datasets show that our method efficiently generates image and video adversarial examples, and more importantly, these adversarial examples have better transferability, therefore being able to simultaneously attack two kinds of representative object detection models: proposal based models like Faster-RCNN and regression based models like SSD.

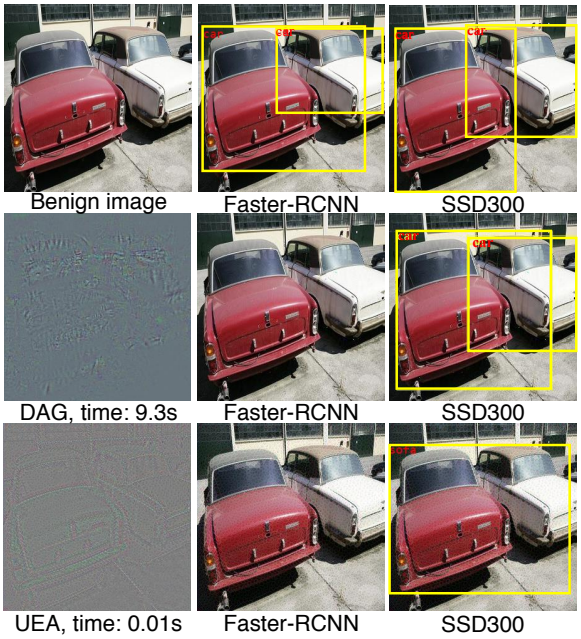

Figure 1: An example of the comparisons between DAG (Dense Adversary Generation) and our UEA (Unified and Efficient Adversary) against proposal and regression based detectors. In the first row, Faster-RCNN and SSD300 detect the correct objects. The second row lists the adversarial examples from DAG. We see it succeeds to attack Faster-RCNN, but fails to attack SSD300. In this third row, neither Faster-RCNN nor SSD300 detects the cars on the adversarial images. Moreover, the UEA's processing time is almost 1000 times faster than DAG for generating an adversarial image.

1 Introduction

Deep learning techniques have achieved great success in various computer vision tasks [Zhu *et al.*, 2017a; Zhu *et al.*, 2017b; Wei *et al.*, 2018]. However, it is also proved that neural networks are vulnerable to adversarial examples [Szegedy *et al.*, 2013], thereby attracting a lot of attention on attacking (e.g., FGSM [Goodfellow *et al.*, 2015; Dong *et al.*, 2018], deepfool [Moosavi-Dezfooli *et al.*, 2016], C&W attack [Carlini and Wagner, 2017]) and defending (e.g., [Raghunathan *et al.*, 2018]) a network. Attacking is beneficial for deeply understanding neural networks [Dong *et al.*, 2017] and motivating more robust solutions [Pang *et al.*, 2018]. Though much work has been done for image classification, more and more methods are presented to attack other tasks, such as face recognition [Sharif *et al.*, 2016], video action recognition [Wei *et al.*, 2019], and the physical-world adversarial attack on road signs [Evtimov *et al.*, 2017].

As the core task in computer vision, object detection for image data has also been attacked. It is known that the current object detection models can be roughly categorized into two classes: proposal based models and regression based models. The various mechanisms make attacking object detection more complex than image classification. [Xie *et al.*, 2017]

†Equal Contributions; *Corresponding Author

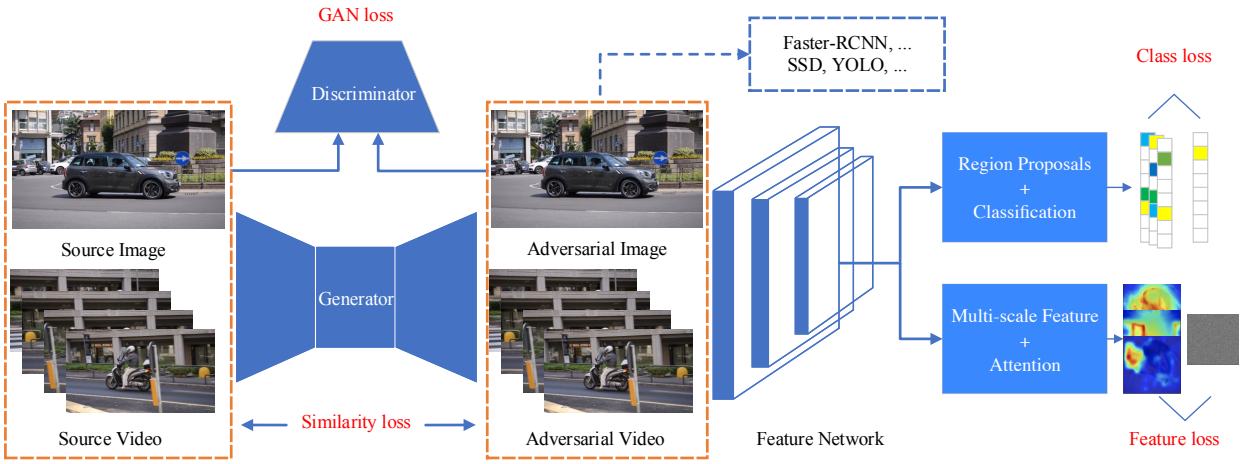

Figure 2: The training framework of Unified and Efficient Adversary (UEA). Besides the GAN loss and similarity loss, we formulate DAG's high-level class loss and our low-level **multi-scale attention feature loss** into GAN framework to jointly train a generator. In the testing phase, the generator is used to output adversarial images or video frames to fool the different classes of object detectors (blue dashed box).

proposes a white-box attacking method for proposal based models: Dense Adversary Generation (DAG). They choose Faster-RCNN [Ren *et al.*, 2017] as the threat model. DAG firstly assigns an adversarial label for each proposal region and then performs iterative gradient back-propagation to misclassify the proposals. The similar methods are also presented in [Chen *et al.*, 2018; Li *et al.*, 2018]. Because regression-based methods don't use region proposals, DAG cannot directly transfer to attack them. That means DAG has weak black-box attacking ability. In addition, DAG is an optimization method, which often needs 150 to 200 iterations to meet the end for an image [Xie *et al.*, 2017]. The high computation cost makes DAG not available for attacking video object detection, which usually considers temporal interactions between adjacent frames [Zhu *et al.*, 2017a] and therefore the most reliable attacking method for video object detection is to pollute all the frames or many key frames in the video.

To address these issues, in this paper, we propose the Unified and Efficient Adversary (UEA) for image and video object detection. "Efficient" specifies that our method is able to quickly generate adversarial images, and thus can efficiently deal with every frame in the video. To this end, we utilize a generative mechanism instead of the optimization procedure. Specifically, we formulate the problem into Generative Adversarial Network (GAN) framework like [Xiao *et al.*, 2018; Poursaeed *et al.*, 2018], and train a generator network to generate adversarial images and key frames. Because the testing step only involves the forward network, the running time is fast. As for "Unified", it means that the proposed adversary has better transferability than DAG, and thus has strong black-box attacking ability. It can not only perform reliable attack to Faster-RCNN like DAG, but also effectively attack regression based detectors. We observe that both the proposal and regression based detectors utilize feature networks as their backends. For examples, Faster-RCNN and SSD [Liu *et al.*, 2016] use the same VGG16 [Simonyan and Zisserman, 2015]. If we manipulate the features maps in Faster-RCNN, the generated adversarial examples will also make SSD fail to detect objects. This idea is implemented as a multi-scale attention feature loss in our paper, i.e., manipulating the feature maps from multiple layers. To fool detectors, only the regions of foreground objects need perturbing. Therefore, an attention weight is integrated into the feature loss to manipulate the feature subregions. The usage of attention weight also improves the imperceptibility of generated adversarial examples because the number of perturbed pixels is limited. In the viewpoint of DNNs' depth, DAG's class loss is applied on the high-level softmax layer, and attention feature loss is performed on the low-level backend layer. Besides class loss, UEA incorporates an additional feature loss to get the strong transferability, which is reasonable. Figure 1 gives an example of UEA, and Figure 2 illustrates the overall framework.

In summary, this paper has the following contributions:

- We propose the Unified and Efficient Adversary (UEA) for attacking image and video detection. To the best of our knowledge, UEA is the first attacking method that can not only efficiently deal with both images and videos, but also simultaneously fool the proposal based detectors and regression based detectors.
- We propose a multi-scale attention feature loss to enhance the UEA's black-box attacking ability. Furthermore, we formulate the existing high-level class loss and the proposed low-level feature loss within GAN framework to jointly train a better generator.

The rest of this paper is organized as follows. In Section 2, we briefly review the related work. We present the proposed Unified and Efficient Adversary framework in Section 3. Section 4 reports all experimental results. Finally, we summarize the conclusion in Section 5.

2 Related Work

The related work comes from two aspects: image and video object detection and adversarial attack for object detection.

2.1 Image and Video Object Detection

Object detection is an area where deep learning has shown its great power. Currently, the dominant image object detection models can be roughly categorized into two classes: proposal based models and regression based models. The former class typically contains R-CNN [Girshick *et al.*, 2016], Faster-RCNN [Ren *et al.*, 2017], Mask-RCNN [He *et al.*, 2017], etc. These kinds of methods use a two-step procedure. They firstly detect proposal regions, and then classify them to output the final detected results. The latter class is represented by YOLO [Redmon *et al.*, 2016] and SSD [Liu *et al.*, 2016]. They regard the detection task as the regression process, and directly predict the coordinates of bounding boxes. Compared with the image scenario, video object detection incorporates temporal interactions between adjacent frames into the procedure. They usually apply the existing image detector on the selected key frames, and then propagate the bounding boxes via temporal interactions [Zhu *et al.*, 2017a; Zhu *et al.*, 2017b]. Therefore, image object detection forms the basis of the video object detection. In this paper, we aim to present a unified method that can attack both the image and video detectors.

2.2 Adversarial Attack for Object Detection

Currently, adversarial attacks for the object detection are rare. The first method is proposed by [Xie *et al.*, 2017], named DAG. They firstly assign an adversarial label for each proposal region and then perform iterative gradient back-propagation to misclassify the proposals. DAG is based on the optimization, and is time consuming, it needs many iterations to accomplish an adversarial image. [Chen *et al.*, 2018; Li *et al.*, 2018] present the similar idea. In addition, [Bose and Aarabi, 2018] tries to attack the face detector. But their threat model is also based on proposal based detectors (Faster-RCNN). All these works attack the proposal based object detectors, and they are all based on the the optimization manner. A unified adversary, which can simultaneously attack both the proposal based and regression based detectors, is absent. In this paper, we aim to fill in this gap, and present a unified method that can attack both the detectors.

3 Methodology

In this section, we introduce the details of UEA.

3.1 Problem Definition

Given an image I, our goal is to generate its corresponding adversarial image $\hat{I}$. We hope that $\hat{I}$ can attack the object detector Dt. For a ground-truth object (B_i, C_i) on I, where B_i is the bounding box, and C_i is the label. Suppose the object detector Dt succeeds to detect this object and outputs (b_i, c_i), where the IOU between B_i and b_i is more than 0.5, and $C_i = c_i$. We let $(\hat{b}_i, \hat{c}_i)$ denote the detected result of this object on the adversarial image $\hat{I}$ (Note that $\hat{b}_i$ may be empty, which represents Dt doesn't detect this object). If the IOU between $\hat{b}_i$ and B_i is less than 0.5 or $\hat{c}_i \neq C_i$, we can say the object detector Dt is successfully attacked or fooled. In order to measure the performance of attacking methods, we will compute the detection accuracy using mAP (mean Average Precision) on the entire dataset, and check the mAP drop after attacks. For videos, we regard the key frames in a video as images, and perform the same operation. We expect the adversarial video can also fool the state-of-the-art video detection models. The Dt is based on proposals or regression.

3.2 Unified and Efficient Adversary

In this section, we introduce the technical details of UEA. Overall, we utilize a generative mechanism to accomplish this task. Specifically, we formulate our problem into the conditional GAN framework. The objective of the conditional GAN can be expressed as:

$$\mathcal{L}_{cGAN}(\mathcal{G}, \mathcal{D}) = \mathbb{E}_I[\log \mathcal{D}(I)] + \mathbb{E}_I[\log(1 - \mathcal{D}(\mathcal{G}(I)))], \quad (1)$$

where $\mathcal{G}$ is the generator to compute adversarial examples, and $\mathcal{D}$ is the discriminator to distinguish the adversarial examples from the clean images. Because adversarial examples are defined as close as as possible with original examples [Szegedy *et al.*, 2013], we input the original images (or frames) and adversarial images (or frames) to the discriminator to compute GAN loss in Eq.(1). In addition, an L_2 loss between the clean images (or frames) and adversarial images (or frames) is applied to measure their similarity:

$$\mathcal{L}_{L_2}(\mathcal{G}) = \mathbb{E}_I[||I - \mathcal{G}(I)||_2]. \quad (2)$$

After training the generator based on GAN framework, we use this generator to generate adversarial examples for testing images and videos. The adversarial examples are then fed into object detectors to accomplish the attacking task.

3.3 Network Architecture

Essentially, the adversarial example generation can be formulated into an image-to-image translation problem. The clean images or frames are input, and the adversarial images or frames are output. Therefore, we can refer to the training manner of pix2pix [Isola *et al.*, 2017]. In this paper, we utilize the network architecture in [Xiao *et al.*, 2018] for ImageNet images, that is the first framework to generate adversarial examples using a pix2pix adversarial generative network. The generator is an encoder-decoder network with 19 components. The discriminator is similar to ResNet-32 for CIFAR-10 and MNIST. Please refer to [Xiao *et al.*, 2018] for the detailed structure of the generator and discriminator.

3.4 Loss Functions

To simultaneously attack the current two kinds of object detectors, we need additional loss functions on the basis of Eq.(1) and Eq.(2). To fool Faster-RCNN detector, DAG [Xie *et al.*, 2017] uses a misclassify loss to make the predictions of all proposal regions go wrong. We also integrate this loss. The class loss function is defined as follows:

$$\mathcal{L}_{DAG}(\mathcal{G}) = \mathbb{E}_I[\sum_{n=1}^{N}[f_{l_n}(\mathbf{X}, t_n) - f_{\hat{l}_n}(\mathbf{X}, t_n)]], \quad (3)$$

where $\mathbf{X}$ is the extracted feature map from the feature network of Faster-RCNN on I, and $\tau = \{t_1, t_2, ..., t_N\}$ is the set of all proposal regions on $\mathbf{X}$. t_n is the n-th proposal region from the

Region Proposal Network (RPN). l_n is the ground-truth label of t_n, and $\hat{l}_n$ is the wrong label randomly sampled from other incorrect classes. $f_{l_n}(\mathbf{X}, t_n) \in \mathbb{R}^C$ denotes the classification score vector (before softmax normalization) on the n-th proposal region. In the experiments, we pick the proposals with score $\geq$0.7 to form $\tau = \{t_1, t_2, ..., t_N\}$.

DAG loss function is specially designed for attacking Faster-RCNN, therefore its transferability to other kinds of models is weak. To address this issue, we propose the following multi-scale attention feature loss:

$$\mathcal{L}_{Fea}(\mathcal{G}) = \mathbb{E}_I[\sum_{m=1}^{M} ||\mathbf{A}_m \circ (\mathbf{X}_m - \mathbf{R}_m)||_2], \qquad (4)$$

where $\mathbf{X}_m$ is the extracted feature map in the m-th layer of the feature network. $\mathbf{R}_m$ is a randomly predefined feature map, and is fixed during training. To fool detectors, only the regions of foreground objects need perturbing. We use the attention weight $\mathbf{A}_m$ to measure the objects in $\mathbf{X}_m$. $\mathbf{A}_m$ is computed based on the region proposals of RPN. We let s_n denote the score of region proposal t_n. For each pixel in the original image, we collect all the region proposals covering this pixel, and compute the sum **S** of these proposals' scores s_n, and then divide **S** by the number of proposals N to obtain the attention weight in the original image. Finally, $\mathbf{A}_m$ is obtained by mapping the original attention weight to the m-th feature layer. For the pixels within objects, their weights will have large values and vice verse. $\circ$ is the Hadamard product between two matrices. By making $\mathbf{X}_m$ as close as $\mathbf{R}_m$, Eq.(4) enforces the attention feature maps to be random permutation, and thus manipulates the feature patterns of foreground objects. $\mathbf{R}_m$ can also be replaced by other feature maps different from $\mathbf{X}_m$. In the experiments, we choose the Relu layer after conv3-3 and the Relu layer after conv4-2 in VGG16 to manipulate their feature maps. To compute $\mathbf{A}_m$, we use the top 300 region proposals according to their scores.

Finally, our full objective can be expressed as:

$$\mathcal{L} = \mathcal{L}_{cGAN} + \alpha\mathcal{L}_{L_2} + \beta\mathcal{L}_{DAG} + \epsilon\mathcal{L}_{Fea}, \qquad (5)$$

where α, β, ϵ are the relative importance of each objective. We set $\alpha = 0.05, \beta = 1$. For ϵ, we set 1×10^{-4} and 2×10^{-4} for the selected two layers, respectively. $\mathcal{G}$ and $\mathcal{D}$ are obtained by solving the minmax game $argmin_{\mathcal{G}} max_{\mathcal{D}} \mathcal{L}$. To optimize our networks under Eq.(5), we follow the standard approach from [Isola *et al.*, 2017] and apply the Adam solver [Kingma and Ba, 2014]. The best weights are obtained after 6 epochs.

4 Experiments

4.1 Datasets

For image detection, we use the training dataset of PASCAL VOC 2007 with totally 5011 images to train the adversarial generator. They are categorized into 20 classes. In testing, we use the PASCAL VOC 2007 testing set with 4952 images.

For video detection, we use ImageNet VID dataset. There are 759 video snippets for training set, and 138 for testing set. The frame rate is 25 or 30 fps for most snippets. There are 30 object categories, which are a subset of the categories in the ImageNet dataset.

4.2 Metrics

We use two metrics: attacking performance against object detectors; the generating time for adversarial examples.

Fooling Rate: to test the fooling rate of different attacking methods, we use the mAP drop (mean Average Precision). The mAP is usually to evaluate the recognition accuracy of object detectors both for image and video data. If the adversary is strong, detectors will achieve a lower mAP on adversarial examples than clean examples. The reducing error can be used to measure the attacking methods.

Time: to tackle with video data, the time for generating adversarial examples is important. In the experiments, we report the processing time for each image (frame) against different attacking methods.

4.3 Threat Models

For image detection, our goal is to simultaneously attack the proposal based detectors and regression based detectors. We select two representative methods: Faster-RCNN and SSD300. There are a lot of implementation codes for them. Here we use the Simple Faster-RCNN and torchCV SSD300. We retrain their models on PASCAL VOC training datasets. Specifically, Faster-RCNN is trained on the PASCAL VOC 2007 training dataset, and tested on the PASCAL VOC 2007 testing set. The detection accuracy (mAP) reaches 0.70. SSD300 is trained on the hybrid dataset consisting of PASCAL VOC 2007 and 2012 training set, and tested on the PASCAL VOC 2007 testing set. The mAP reaches 0.68.

For video detection, the current video detection methods are based on image detection. They usually perform image detection on key frames, and then propagate the results to other frames [Zhu *et al.*, 2017a; Zhu *et al.*, 2017b]. However, as shown in [Zhu *et al.*, 2017b], the detection accuracy of these efficient methods cannot even outperform the simple dense detection method, that densely runs the image detection on each frame in a video. In [Zhu *et al.*, 2017a], although their method beats dense detection method, they cost more time. If they reduce the processing time, the accuracy also falls below the dense detection. Therefore, we choose the dense detection method as the threat model. We argue if the dense detection method is successfully attacked, the efficient methods will also fail.

Methods	Accuracy (mAP)		Time (s)
	Faster-RCNN	SSD300	
Clean Images	0.70	0.68	\
DAG	0.05	0.64	9.3
UEA	**0.05**	**0.20**	**0.01**

Table 1: The mAP and Time comparisons between DAG and UEA.

http://bvisionweb1.cs.unc.edu/ILSVRC2017/download-videos-1p39.php

https://github.com/chenyuntc/simple-faster-rcnn-pytorch
https://github.com/kuangliu/torchcv

4.4 Results on Image Detection

Comparisons with State-of-the-art Methods

The current state-of-the-art attacking method for image detection is DAG. Therefore, we use DAG as our compared method. For that, we generate adversarial image using DAG and UEA, respectively, and then perform the same Faster-RCNN and SSD300 on the adversarial examples to observe the accuracy drop (compared with the accuracy on clean images). Meanwhile, we also check the generating time (Time). The comparison results are reported in Table 1.

From the table, we see: **(1):** Both DAG and UEA work well on attacking Faster-RCNN detector. They achieve the same 0.65 accuracy drop (0.70-0.05). This is expected because DAG and UEA formulate the same class loss of Eq.(3) into their methods, and they perform the white-box attack against Faster-RCNN. **(2):** DAG cannot attack SSD detector, the accuracy drop is only 0.04 (0.68-0.64). By contrast, UEA obtains a 0.48 accuracy drop (0.68-0.20), which is 12 times larger than DAG. This verifies the weak black-box attacking ability of DAG. Instead, UEA integrates a feature loss to manipulate the shared feature networks between Faster-RCNN and SSD. The feature loss enhances the transferability and black-box attacking ability to other kinds of detectors. Theoretically, UEA is able to attack a large class of object detectors besides SSD and Faster-RCNN, because the majority of object detectors use the feature network. **(3):** As for the generating time of adversarial examples, UEA is almost 1000 times faster than DAG (0.01 vs 9.3). The efficiency is helpful to tackle with video data. Even for the video with 100 frames, UEA will only cost one second to pollute all the frames.

We also evaluate the perceptibility of adversarial examples. Figure 3 gives the comparisons. As an optimization method, DAG is highly relevant with different images. Their perturbations are increasing with the rising iterations. For example, DAG only costs 1 iteration for "cat" image, and the perturbations are imperceptible. But for "motorbike" image, DAG costs 81 iterations, and the perturbations are very obvious (see the regions in red circles). The "cow" and "boat" images have the similar trend (see the regions in red circles). UEA is a generative method. We see the adversarial examples are always imperceptible, and almost the same as the clean images.

Ablation Study of UEA

Now we look into the ablation study of UEA. As introduced in Section 3, UEA utilizes two key loss functions in the training phase. The first is class loss, i.e., Eq.(3), and the second is multi-scale attention feature loss, i.e., Eq.(4). We study the function of each loss, and report the results for each category detection in Figure 4. In this figure, Y-axis is the mAP, X-axis is the category index in PASCAl VOC. In sequence, they denote "Airplane", "Bicycle", "Bird", "Bottle", "Bus", "Car", "Cat", "Chair", "Cow", "Table", "Dog", "Horse", "Motorbike", "Person", "Plant", "Sheep", "Sofa", "Train", "TV". Blue curve denotes the class loss. Red curve denotes the feature loss. Black curve denotes the hybrid loss, which is the full version of UEA with both class loss and feature loss. From the figure, we see that "class loss" works well on Faster-RCNN, but shows the limited attacking ability on SSD300. After adding the proposed "feature loss", UEA has the similar attacking performance with "class loss" on Faster-RCNN, but shows stronger attacking ability on SSD300. These results demonstrate that hybrid the high-level class loss and low-level feature loss is a reasonable choice.

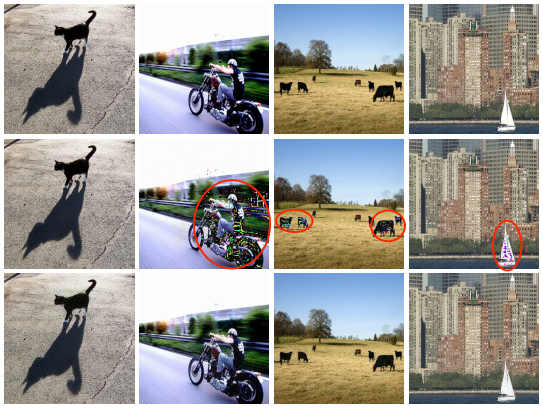

Figure 3: The perceptibility comparison of adversarial images. The first row is clean images. The second row is output by DAG (the iteration is 1, 81, 133, 41, respectively). The third row is our output.

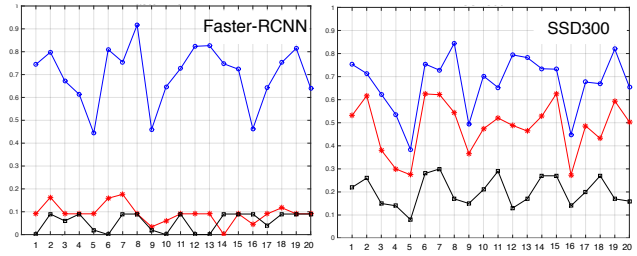

Figure 4: The ablation study of UEA for each category detection.

Qualitative Comparisons

We give some qualitative comparisons between DAG and UEA in Figure 5. From the figure, we see both Faster-RCNN and SSD300 work well on the clean images, and detect the correct bounding boxes and labels. For DAG, it succeeds to attack Faster-RCNN (see the sixth row where Faster-RCNN doesn't detect any object on two images and predicts wrong labels on three images). However, SSD300 still works well on the adversarial examples generated by DAG. For UEA, Faster-RCNN cannot detect any bounding box on five adversarial examples, and SSD300 detects wrong objects on two images and zero detection on three images.

To better show the intrinsic mechanism of UEA, we visualize the feature maps extracted from adversarial examples via DAG and UEA, respectively. Because Faster-RCNN and SSD300 utilize the same VGG16 as their feature network, we select the feature maps extracted on conv4 layer and visualize them using the method in [Zeiler and Fergus, 2014]. From Figure 6, we see that the feature maps via UEA have been manipulated. Therefore, the Region Proposal Network within Faster-RCNN cannot output the available proposal regions, and thus Faster-RCNN doesn't detect any bounding box. For SSD300, the manipulated features make the regression operation not work, leading to wrong or vacant predictions.

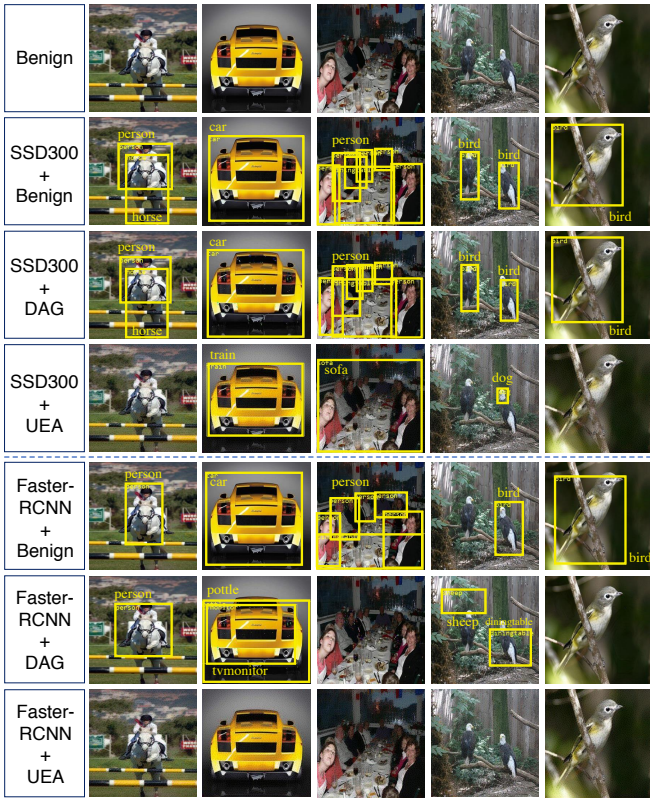

Figure 5: The qualitative comparisons between DAG and UEA versus Faster-RCNN and SSD300. Please see the texts for details.

4.5 Results on Video Detection

In this section, we report the results on video object detection. We here use the ImageNet VID dataset. As discussed in section 4.3, we attack the dense frame detection methods. Specifically, we train Faster-RCNN and SSD300 on ImageNet VID dataset, and then run the detectors on each frame in the testing video. We believe that if the dense frame detection method can be successfully attacked, other efficient methods will be also fooled. More qualitative attacking results can be found in https://sites.google.com/view/ueaattack/home

Table 2 shows the quantitative attacking performance of UEA on ImageNet VID. Specifically, we train Faster-RCNN and SSD300 on the training set of ImageNet VID, and run the trained detectors on the testing set. In addition, we use UEA to generate the corresponding adversarial videos for the testing set of ImageNet VID, and then run the same detectors. In Table 2, we see UEA achieves 0.40 mAP drop for Faster-RCNN, and 0.44 mAP drop for SSD300, which shows UEA achieves a good attacking performance in the video data.

We here use the VGG16 based Faster-RCNN and SSD300. [Zhu *et al.*, 2017b] shows that if we use ResNet 101 as the backbone network, and replace Faster-RCNN with FCN [Dai *et al.*, 2016] as the object detector, the original mAP will reach 0.73. Because our paper aims at measuring the attacking ability of UEA, rather than the detecting performance, the mAP drop is the key metric, rather than mAP. Therefore, we here don't use ResNet 101+FCN. Similarly, we also don't use the SSD500, although it has better detection than SSD300. The current mAP drop has verified the powerful attacking ability of UEA both against the proposal based detector (Faster-RCNN) and regression based detector (SSD300). We believe that if we use the advanced object detectors, the mAP drop will also improve.

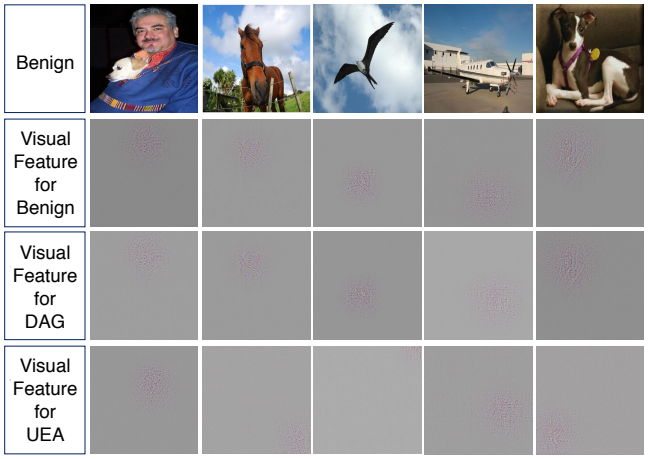

Figure 6: The feature visualization of adversarial examples via DAG and UEA, respectively. Please see the texts for details.

Methods	Accuracy (mAP)		Time (s)
	Faster-RCNN	SSD300	
Clean Videos	0.43	0.50	\
UEA	0.03	0.06	0.3s
mAP drop	**0.40**	**0.44**	\

Table 2: The attacking performance of UEA on video detection.

5 Conclusion

In this paper, we proposed the Unified and Efficient Adversary (UEA). UEA was able to efficiently generate adversarial examples, and its processing time was 1000 times faster than the current attacking methods. Therefore, UEA could deal with not only image data, but also video data. More importantly, UEA had better transferability than the existing attacking methods, and thus, it could meanwhile attack the current two kinds of representative object detectors. Experiments conducted on PASCAL VOC and ImageNet VID verified the effectiveness and efficiency of UEA.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (No.2017YFA0700904, 2018YFB0803701), the NSFC Projects (Nos. 61806109, 61673241, 61620106010, 61621136008, 61332007, U1636214), Project funded by China Postdoctoral Science Foundation (No.2018M641360, 2019T120094), the MIIT Grant of Int. Man. Comp. Stan (No. 2016ZXFB00001), Tsinghua Tiangong Institute for Intelligent Computing, the NVIDIA NVAIL Program and a Project from Siemens.

References

[Bose and Aarabi, 2018] Avishek Joey Bose and Parham Aarabi. Adversarial attacks on face detectors using neural net based constrained optimization. *IEEE MMSP*, 2018.

[Carlini and Wagner, 2017] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE S&P*, pages 39–57, 2017.

[Chen *et al.*, 2018] 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.

[Dai *et al.*, 2016] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NeurIPS*, pages 379–387, 2016.

[Dong *et al.*, 2017] Yinpeng Dong, Hang Su, Jun Zhu, and Fan Bao. Towards interpretable deep neural networks by leveraging adversarial examples. *arXiv preprint arXiv:1708.05493*, 2017.

[Dong *et al.*, 2018] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, pages 9185–9193, 2018.

[Evtimov *et al.*, 2017] Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. Robust physical-world attacks on deep learning models. *arXiv preprint arXiv:1707.08945*, 1, 2017.

[Girshick *et al.*, 2016] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE TPAMI*, 38(1):142–158, 2016.

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

[He *et al.*, 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2980–2988, 2017.

[Isola *et al.*, 2017] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *ICCV*, 2017.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Li *et al.*, 2018] Yuezun Li, Daniel Tian, Xiao Bian, Siwei Lyu, et al. Robust adversarial perturbation on deep proposal-based models. *BMVC*, 2018.

[Liu *et al.*, 2016] 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.

[Moosavi-Dezfooli *et al.*, 2016] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, pages 2574–2582, 2016.

[Pang *et al.*, 2018] Tianyu Pang, Chao Du, Yinpeng Dong, and Jun Zhu. Towards robust detection of adversarial examples. In *NeurIPS*, pages 4584–4594, 2018.

[Poursaeed *et al.*, 2018] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *CVPR*, June 2018.

[Raghunathan *et al.*, 2018] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018.

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

[Ren *et al.*, 2017] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE TPAMI*, (6):1137–1149, 2017.

[Sharif *et al.*, 2016] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *CCS*, pages 1528–1540, 2016.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

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

[Wei *et al.*, 2018] Xingxing Wei, Jun Zhu, Sitong Feng, and Hang Su. Video-to-video translation with global temporal consistency. In *ACMMM*, pages 18–25, 2018.

[Wei *et al.*, 2019] Xingxing Wei, Jun Zhu, Yuan Sha, and Hang Su. Sparse adversarial perturbations for videos. In *AAAI*, 2019.

[Xiao *et al.*, 2018] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.

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

[Zeiler and Fergus, 2014] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833, 2014.

[Zhu *et al.*, 2017a] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *ICCV*, volume 3, 2017.

[Zhu *et al.*, 2017b] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *ICCV*, volume 1, page 3, 2017.