1 Experiments on OpenImage Dataset

To further evaluate the threats caused by non-local block, appearing and disappearing attacks are performed on the Faster R-CNN with the non-local block (Faster R-CNN-WN) on Open Images Dataset V4. The object detector is trained on COCO dataset with 80 classes of objects. 24 classes overlapping with the object classes in Open Images Dataset V4 are selected for the following experiments.

1.1 Disappearing attack

Different from the stop sign dataset, which normally just has 1 to 2 stop signs in each image, images in Open Images Dataset V4 usually have a lot of target objects. The adversarial patch has to be placed with a certain distance to all the target objects. The procedure to place Λ is given in algorithm 1. λ is used to control the distance between Λ and the target object used as a reference position and δ is used to control the minimum distance between Λ and other target objects. Images which do not have enough space to place the adversarial patch are not used in this evaluation. All the target classes have at least 100 valid images for this evaluation. α and β are both set as 1.5. δ and λ are set as 1 and 0.5 respectively. For classes with more than 2000 images, only 2000 images are randomly sampled for the experiment. The images in each class are splitted equally as training and testing sets. Table 1. lists the results for disappearing attack. The average attack rate is 69.9%. Fig. 1 shows some resultant images of disappearing attack.

1.2 Appearing attack

The same 24 object classes are used in appearing attack. For each of the class, its appearing attack can be set as any 1 of the other 79 classes. In total, there are 1896 experiments. To reduce the number of experiments, we perform non-target appearing attacks. It means that if a newly detected bounding box which has not being detected before the attack, and has an IOU with the ground truth bounding box less than 0.7, then it is considered as a success, regardless of its class label. The objective function, which is similar as disappearing attack, aims at minimizing the scores for all the background bounding boxes and the scores for all the corresponding class bounding boxes. Thus, objects would be detected in the original background and other objects would be detected at the locations of the corresponding class objects. In this experiment, α and β are both set as 2. Table 2 lists the experimental results and some resultant images are given in Fig. 2

Algorithm 1: Procedure to add Λ to image

Input: I: original image; $b_{1,..,n}$: ground truth bounding boxes for target objects, n is the number of target objects; Λ: adversarial patch;

Output: a flag indicates if I is available and the position of Λ in I

set the flag to be *false* and the position of Λ to be empty ;
shuffle all the bounding boxes; compute the mean height h_m and width w_m of all $b_i, i \in 1, ..., n$; resize Λ to $\alpha h_m \times \beta w_m$;
for *each bouding box* b_i *in* $b_{1,..,n}$ **do**
    place Λ below b_i with distance δh_i
    if *Λ is inside the image and $D_j/(h_j + w_j) > \lambda$, where D_j is the distance between Λ and the bouding box $b_j, j \in 1, .., n, j \neq i$* **then**
        set the flag to be true and save the current postion of Λ ;
        goto final ;
    place Λ above b_i with distance δh_i
    if *Λ is inside the image and $D_j/(h_j + w_j) > \lambda$, where D_j is the distance between Λ and the bouding box $b_j, j \in 1, .., n, j \neq i$* **then**
        set the flag to be true and save the current postion of Λ ;
        goto final ;
    place Λ on the left side of b_i with distance δw_i
    if *Λ is inside the image and $D_j/(h_j + w_j) > \lambda$, where D_j is the distance between Λ and the bouding box $b_j, j \in 1, .., n, j \neq i$* **then**
        set the flag to be true and save the current postion of Λ ;
        goto final ;
    place Λ on the right side b_i with distance $\delta w_i 5$
    if *Λ is inside the image and $D_j/(h_j + w_j) > \lambda$, where D_j is the distance between Λ and the bouding box $b_j, j \in 1, .., n, j \neq i$* **then**
        set the flag to be true and save the current postion of Λ ;
        goto final ;
there is no suitable position to place Λ, set the flag to be *false*, the position of Λ to be 0 ;
final ;
return the flag and Λ position;

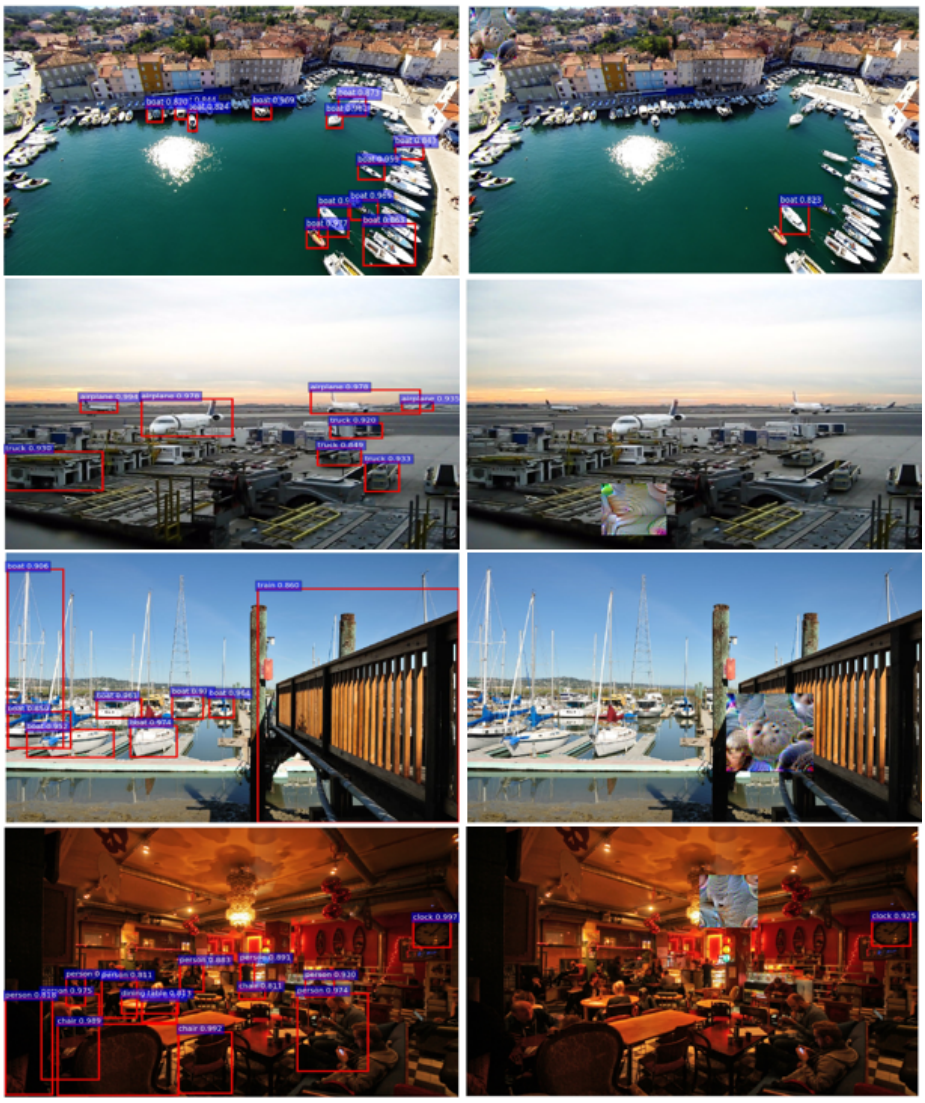

Fig. 1: Disappearing attack results. The first column is the original detection results without the adversarial patches and the second column is the detection results with the adversarial patches.

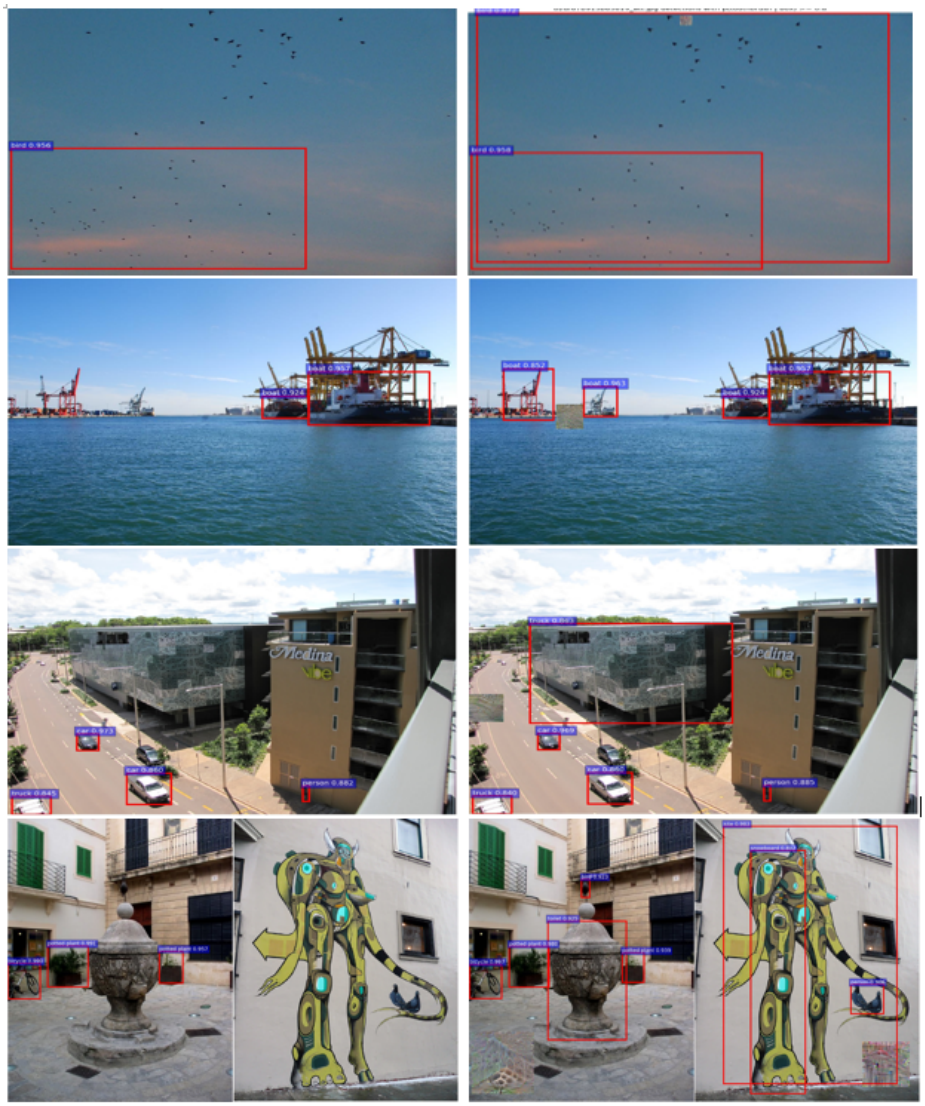

Fig. 2: Appearing attack results. The first column is the original detection results without the adversarial patches and the second column is the detection results with the adversarial patches.

Table 1: The successful disappearing attack rates (%) for different classes.

class	Airplane	Bench	Bottle	Bowl	Bicycle	Bird
detection rate	81.3	76.3	81.7	78.4	71.6	73.4
attack rate	83.2	79.1	61	60.7	55.7	55.9
class	Boat	Bus	Chair	Cake	Car	Cat
detection rate	62.6	83.5	77.5	70.4	79.4	74.1
attack rate	57.5	75.9	80.3	82.6	53.4	96.4
class	Clock	Dog	Fork	Horse	Motorcycle	Person
detection rate	68.0	80.6	71.3	84.1	76.9	78.3
attack rate	18.6	99.3	66.1	89.7	68.3	71.2
class	Tie	Train	Truck	Umbrella	Vase	Wine Glass
detection rate	81.6	67.1	73.3	69.4	78.4	83.0
attack rate	31.6	97	90	95.3	71.6	37.3

Table 2: The successful disappearing attack rates (%) for different classes.

class	Airplane	Bench	Bottle	Bowl	Bicycle	Bird
attack rate	40	29.8	46.1	15.5	34.8	15.3
class	Boat	Bus	Chair	Cake	Car	Cat
attack rate	12	17.6	57	33.3	26.9	28.2
class	Clock	Dog	Fork	Horse	Motorcycle	Person
attack rate	21.1	27.2	16.1	14.1	38	26.7
class	Tie	Train	Truck	Umbrella	Vase	Wine Glass
attack rate	24.7	21.5	25	21.4	17.2	50.6