Supplementary Material for "Exploiting Multi-Object Relationships for Detecting Adversarial Attacks in Complex Scenes"

In the supplementary material, we conduct case study on attacks that can bypass our context consistency checker and benign images that are detected as adversarial (Section 1). We present the implementation details about the baseline models (Feature Squeeze and SCEME) used in the paper.

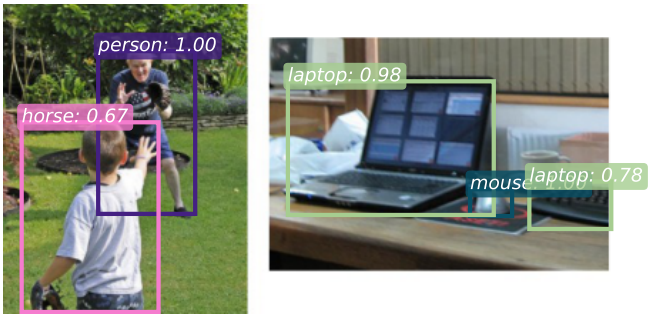

Figure 1: The adversarial examples that our proposed SCENE-BERT model cannot detect.

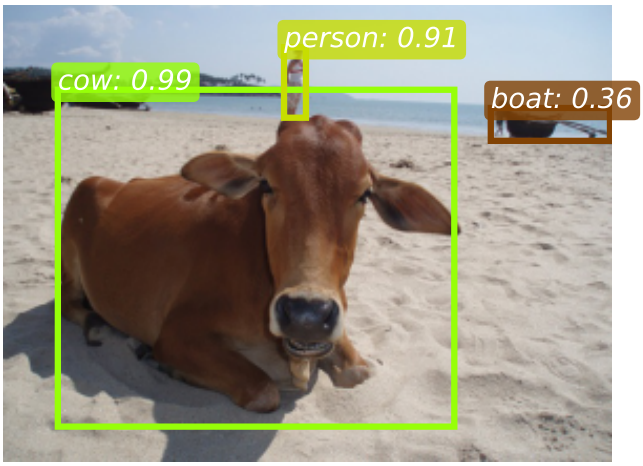

Figure 2: The benign example that our proposed SCENE-BERT model detect as adversarial.

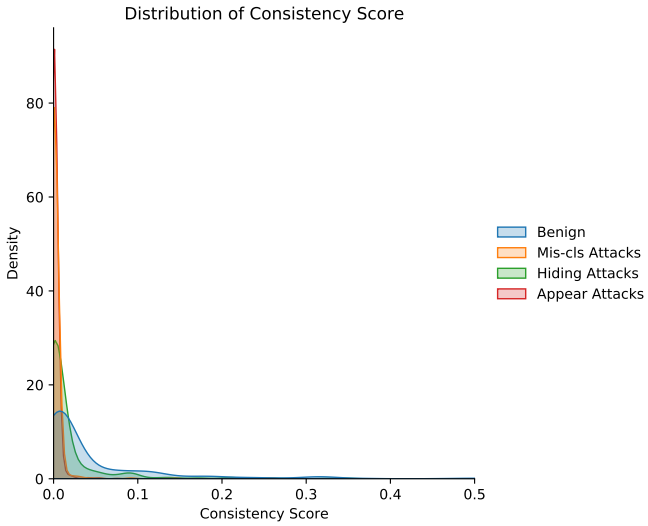

Figure 3: The distribution of consistency score for three types of attacks.

1. Case Study

In this section, we first show (1) attack examples that cannot be detect by our language-based consistency checker SCENE-BERT and (2) benign examples that are reported as adversarial. Then we present the distribution of consistency scores.

False Negatives. For the adversarial image on left hand side of Figure 1, the attack goal is to misclassify the person into a horse. Because the (misclassified) detection result will be described "a person (and) a horse," which is common, our context-consistency-based detector cannot tell the image has been perturbed. For the adversarial image on right hand side of Figure 1, the attack goal is to misclassify the rightmost keyboard into a laptop. Similarly, because it is ordinary for a laptop to co-occur with another laptop, our approach cannot detect the image as perturbed.

False Positives. There are also benign examples that are detected as adversarial by SCENE-BERT. Figure 2 shows an example. In this case, a person lying on top of a cow is common, but a boat co-occur with cow is rare in the training

set or has never been seen before. In this case, though the object boat is indeed an out-of-context anomaly, it is not an attack.

Density Graph. To understand how common such cases are, we plotted the distribution of consistency score for the three types of attacks and the benign cases in Figure 3. As we can see, the consistency score for misclassification appear attacks mainly concentrate at around 0, while the majority of the benign images have higher consistency score. Hiding attacks are hard to detect because hiding attacks usually do not violate context-consistency (e.g., hide one person from a group of person will not cause the context change too much). However, hiding attacks are still detectable in subtle cases. For instance, a watch should be wear by a person or located on a table, if the person are hidden and not table detected, then the consistency can be considered attacked.

These false negatives and false positives are caused by a fundamental limitation of our consistency-based attack detection approach—if the attack itself is context-aware, then we cannot use context-consistency to detect such attacks; on the other hand, if a benign case has never been seen before, then we may also report it as attacks. However, we argue that (1) false positives can be reduced by extending the training set (e.g., by using natural language datasets), (2) as context imposes additional constraints, constructing context-aware attacks are likely to be more expensive; and (3) more importantly, context-consistent attacks may cause be able to lead to dire consequences (e.g., misclassifying STOP sign to YIELD sign may not lead to traffic accidents).

2. Implementation Details

2.1. Feature Squeeze

In this subsection, we explain how the baseline Feature Squeeze is implemented. Algorithm 1 shows the algorithm. $\mathbf{Img_O}$ denotes the original image, $\mathbf{Img_Q}$ defined in line 1 denotes the quantized image after squeezing. $\mathbf{PR_O}$ defined in line 2 denotes the prediction result for the original image from object detector $g(\cdot)$, while $\mathbf{PR_{FS}}$ defined in line 3 denotes the prediction result of quantized image. R_O and R_{FS} defined in line 7, 9 denote one region from the prediction result for original image and quantized image respectively. Note that we take the highest distance among all regions as a represent to the distance of whole image. Furthermore, for each region, we take the lowest distance calculating from all its overlapped as the distance for the queried region. Under the consideration that we usually take the category with highest confidence score for the regions dumped from object detector, we manually set rule, i.e. the distance is 1 for regions with different predicted categories.

Algorithm 1: Calculate the distance using Feature Squeeze of an image.

Input : Image to be tested $\mathbf{Img_O}$,
the quantilize function $f(\cdot)$,
the object detector $g(\cdot)$
Output: Distance d

1 $\mathbf{Img_Q} = f(\mathbf{Img_O})$
2 $\mathbf{PR_O} = g(\mathbf{Img_O})$
3 $\mathbf{PR_{FS}} = g(\mathbf{Img_Q})$
4 $d = 0$
5 $n =$ length of PRFS
6 **for** $i = 1$ *to* n **do**
7 $\quad R_{FS} = \mathbf{PR_{FS}}[\mathbf{i}]$
8 $\quad minDistance = 1$
9 $\quad$ **for** R_O *in* $getOverlap(R_{FS}, \mathbf{PR_O})$ **do**
10 $\quad\quad distance = getDistance(R_{FS}, R_O)$
11 $\quad\quad minDistance = \min(distance, minDistance)$
12 $\quad$ **end**
13 $\quad d = \max(minDistance, d)$
14 **end**
15 **if** R_O *in* $\mathbf{PR_O}$ *overlap with nothing in* $\mathbf{PR_{FS}}$ **then**
16 $\quad d = 1$
17 **end**
18 **return** d

2.2. SCEME

In this subsection, we explain how the baseline SCEME model is implemented. Algorithm 2 illustrates how we adapted the original SCEME, which works at region proposal level, to make it work at whole image level. $\mathbf{PR}$ defined in line 2 denotes the prediction results. CP defined in line 6 denotes the Context Profile extracted from intermediate layer of F-RCNN which will be passed to SCEME model to generate a reconstruction error. We take the highest reconstruction error as the reconstruction error of the whole image.

Algorithm 2: Calculate the SCEME reconstruction error at image level.

Input : Image to be tested $\mathbf{Img}$,
the F-RCNN object detector $f(\cdot)$,
the trained SCEME function $g(\cdot, \cdot)$,
Output: Reconstruction Error e

1 $e = 0$
2 $\mathbf{PR} = f(\mathbf{Img})$
3 $n =$ length of $\mathbf{PR}$
4 **for** $i = 1$ *to* n **do**
5 $\quad Category, CP = \mathbf{PR}[i]$
6 $\quad rec = g(Category, CP)$
7 $\quad e = \max(e, rec)$
8 **end**
9 **return** e

Note that because the categories in MS COCO dataset are biased and SCEME requires on auto-encoder for each category, it cannot be trained well on categories that rarely occur. For this reason, we did not measure SCEME's performance on the MS COCO dataset.

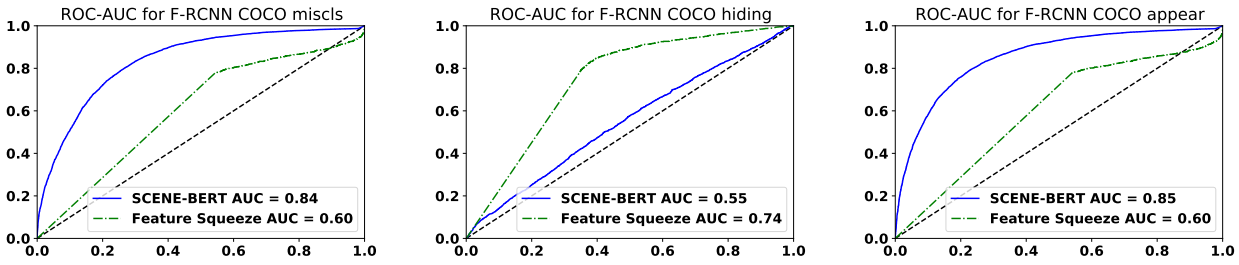

Figure 4: ROC-AUC for F-RCNN on COCO

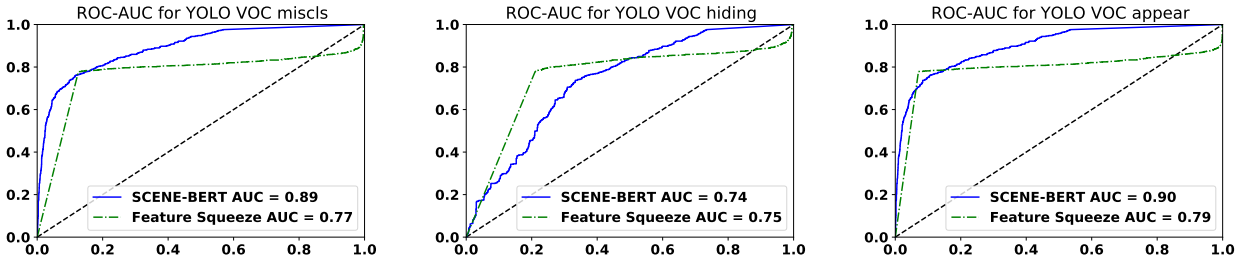

Figure 5: ROC-AUC for YOLO on VOC

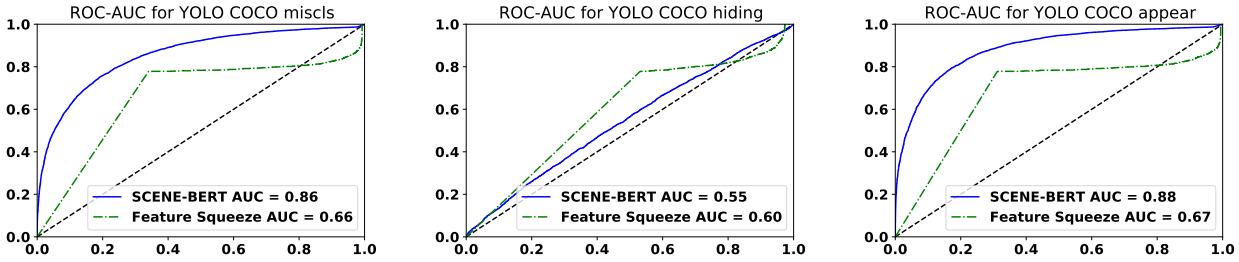

Figure 6: ROC-AUC for YOLO on COCO

3. Additional Results

We plot the ROC-AUC curves for F-RCNN on COCO, YOLO on VOC, and YOLO on COCO in Figure 4, Figure 5, and Figure 6.