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

Mingjun Yin, Shasha Li, Zikui Cai, Chengyu Song,
M. Salman Asif, Amit K. Roy-Chowdhury, and Srikanth V. Krishnamurthy
University of California, Riverside, USA
{myin013, sli057, zcai032}@ucr.edu, csong@cs.ucr.edu,
{sasif, amitrc}@ece.ucr.edu, krish@cs.ucr.edu

Abstract

Vision systems that deploy Deep Neural Networks (DNNs) are known to be vulnerable to adversarial examples. Recent research has shown that checking the intrinsic consistencies in the input data is a promising way to detect adversarial attacks (e.g., by checking the object co-occurrence relationships in complex scenes). However, existing approaches are tied to specific models and do not offer generalizability. Motivated by the observation that language descriptions of natural scene images have already captured the object co-occurrence relationships that can be learned by a language model, we develop a novel approach to perform context consistency checks using such language models. The distinguishing aspect of our approach is that it is independent of the deployed object detector and yet offers very high accuracy in terms of detecting adversarial examples in practical scenes with multiple objects. Experiments on the PASCAL VOC and MS COCO datasets show that our method can outperform state-of-the-art methods in detecting adversarial attacks.

1. Introduction

Deep neural networks (DNNs) are widely used in vision tasks such as object detection and classification, for their ability to achieve state-of-the art (SOTA) performance in such tasks. DNN-based vision systems are also known to be vulnerable to adversarial examples [13, 38, 14, 20, 4, 1, 31]; specifically, it is possible to add (quasi-)imperceptible perturbations that can cause DNN-based vision systems to output incorrect results, while projecting high confidence with regard to the results. For example, adversarial examples can misclassify STOP signs to speed limit signs [10] and a school bus to an ostrich [38].

One promising defense strategy proposed recently is to capture the intrinsic dependencies within the input data, and to check for violations of such dependencies to detect adversarial examples. For instance, in scene images with multiple objects, the intrinsic relationships between objects, commonly known as the *context* of the scene, can be used to detect adversarial attacks [21]. Similarly, the dependencies between video frames can be used to detect adversarial frames in video classification [18, 41]. To illustrate, let us consider the STOP sign attack as an example. A STOP sign is a part of a road intersection scene wherein it typically co-exists with a stop line and/or a street name sign; in contrast, a speed limit sign is rarely, if ever, seen at intersections and thus does not co-exist with the latter objects.

While context has been used extensively for object recognition problems and scene understanding, there is little work with respect to detection of adversarial attacks using context. In our previous work [21], we proposed modeling context as a fully connected graph, where each node is an object proposal from a Region Proposal Network (RPN), and edges encode how other regions (including the background and the whole scene) affect the current node in its feature space. Then we train a bank of auto-encoders (each corresponds to a category of objects) to check for consistency with respect to the distribution of context features. While this approach performs well, it is deeply coupled with Faster R-CNN [36] and cannot be applied to single-stage detectors like YOLO [35]; besides, it requires retraining when there is any change to the Faster R-CNN model (e.g., when switching to another CNN model). In summary, while prior approaches have tried to utilize context to detect adversarial attacks, they do so in a way that intricately ties the context to the model in use, which limits their applicability.

In this paper, we propose a novel *model-agnostic* adversarial attack detector based on object co-occurrence. Our observation is that the language description of a natural scene image (i.e., of the output of an object detection network) can readily capture the dependencies between ob-

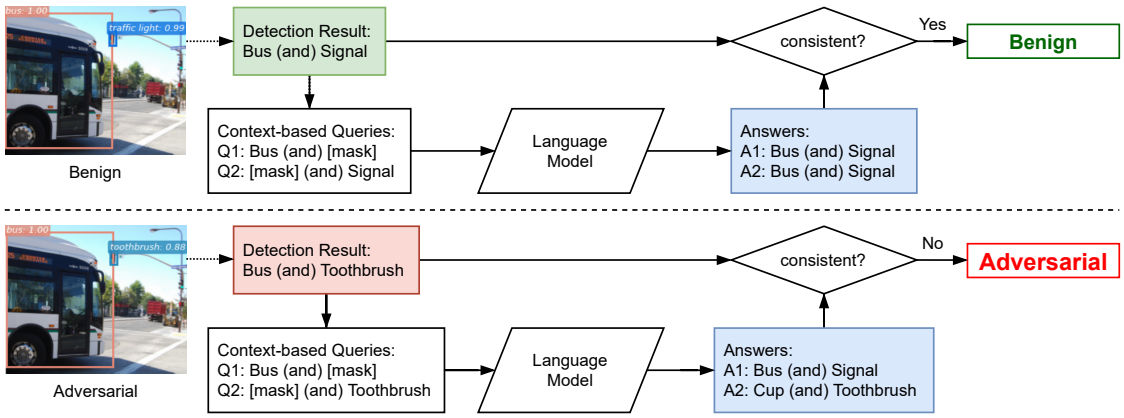


Figure 1: High-level idea of how our language model-based context consistency check works. First, we use a language model to learn the object co-occurrence context (e.g., bus and signal in the example) from descriptions of scene images. At the test time, we mask off detected objects in the scene description and ask the language model to predict the object based on context (i.e., other objects). By measuring the consistency between the detection results and the prediction results, we assess if the input scene image is adversarial or not.

jects. We exploit recent advances in natural language models to learn the dependencies between objects based on co-occurrence and to detect adversarial attacks as violations of the learned context model. Figure 1 depicts the high-level idea of our approach. Given an unknown scene image, we first encode the output of an object detection network into a sentence describing the object co-occurrence relationships (e.g., "bus and signals"). Then we use a trained language model to predict each detected object instance *purely based on the context*. Finally, we evaluate the context consistency of the scene image by comparing the language model prediction the and the detection results. If the results are different, we conclude that the input image is adversarial.

The key contributions of our work are as follows.

- To the best of our knowledge we are the first to propose a *model-agnostic*, context-consistency based approach to detect adversarial perturbations against object detectors.
- We design and realize a language-based model to learn the object co-occurrence relationships in complex scenes, which serves as our novel context model to detect adversarial attacks.
- We conduct extensive experiments with three different types of adversarial attacks (misclassification, hiding, and appearing) on two large-scale datasets - PASCAL VOC [9] and Microsoft COCO [24]. Our method yields high detection performance in all the test cases; the ROC-AUC is over 0.72 in most cases, which is 12-69% higher than a state-of-the-art attack detection method [44] that does not use context and is comparable to (only 5% worse) previous context-inconsistency-based adversarial attack detection approach [21] that is model-dependent (tightly coupled with the Faster R-CNN architecture thus cannot be applied to other architectures, like YOLO).

2. Related Work

In this section, we review closely related work.

Object Detection seeks to locate and classify object instances in images or videos. It is a domain that has been extensively studied [36, 26, 35, 23]. Faster R-CNN [36] and YOLO [35] are two state-of-the-art DNN-based object detectors that we considered in this work. F-RCNN uses a two-stage approach where the first stage proposes bounding boxes and the second stage performs classification. YOLO takes a single pass design aiming to reducing the computation complexity and improving detection speed.

Context-aware Object Detection aims to exploit context information to boost the performance of object detection [34, 39, 8, 2]. Earlier approaches incorporate context (object co-occurrence) information as a post-processing step to re-score objects detected by DNN-based object detectors [12, 6, 33]. Recent work has also proposed incorporating context as part of the DNN using recurrent units [28] or neural attention models [17]. Our method takes the post-processing style for (a) the ease of training, and (b) integrating with multiple different object detectors.

Scene Understanding and Caption Generation study the problem of generating natural language descriptions for scene images [43, 29, 37, 15, 47, 45]. Besides recognizing objects in a scene, a description generator also needs to detect the relationships or interactions between objects (e.g., "man riding a horse"). Although descriptions generated by these systems (e.g., the scene graph [19]) contain richer contextual information and are more discriminative, because predicting correct relationships is a much harder problem than object recognition, existing approaches are not robust enough (i.e., their performance over benign images is not very accurate yet). For this reason, we opt for the simpler context graph purely based on object co-occurrence where there is only one relationship between objects—they

co-occur. This context can also be described with a simpler language, which is easier to model as well.

Adversarial Attacks against DNNs are slightly perturbed inputs that can cause a DNN to misbehave [13]. In the visual domain, the perturbations are usually (quasi-) imperceptible noises, but can also be small patches that can be physically applied to a target object [10]. In whitebox settings, adversarial attacks can be generated using gradient-guided optimizations [38, 14, 20, 4, 1, 31]. Because our approach takes the prediction results (i.e., labels) as inputs, different attack methods (i.e., how labels are misclassified) will not affect the attack detection results; so, we only use one attack method in our evaluation.

Adversarial Attacks against Object Detectors has received less attention than those against image classifiers. Most related work focuses on physically realizable attacks [10, 5, 46], especially in the domain of autonomous driving vehicles [3]. A key difference from attacking an image classifier is that there are two additional types of attacks viable against object detectors: *hiding* and *appearing* attack [5, 10]. Since our attack detection method takes the output of an object detector as input, it is not sensitive to whether the attack is physical or digital. Therefore, we only evaluate our approach against digital attacks.

Detectors of Adversarial Attacks aim to distinguish adversarial images from benign ones. Statistics based detection methods rely on different distributions in the feature space between clean images and perturbed ones [16, 11, 25] to detect adversarial attacks. Another approach is to transform the input and compare the output of the DNN over the original input with the output over the transformed input; a large inconsistency usually indicates the input is adversarial [44, 22]. For example, Feature Squeeze [44] is a state-of-the-art method that aims to remove useless features from the input space (e.g., by reducing the bit depth of pixels and smoothening surrounding pixels). We compare with this method in our experiments.

Detecting Adversarial Attacks using Context is a promising defense strategy explored by recent work. Xiao et al. [42] propose a detection method in the task of segmentation based on spatial consistency (i.e., how the prediction result of a pixel differs from surrounding pixels). Xiao et al. [41] also proposed using temporal consistency to detect adversarial frames in video clips. Ma et al., found that the correlation between audio and video can be use to detect adversarial attacks [30]. The closest work to our approach is our earlier work [21], in which we proposed using an object's context profile, which captures four types of relationships among region proposals (spatial context, object-object context, object-background context, and object-scene context) to detect adversarial perturbations. Since the context profile used by our previous work is extracted from the internal layers of the object detection network, it is tightly coupled with the object detector. The approach presented in this work is model-agnostic and thus, does not require expensive retraining to support new object detectors.

3. Methodology

In this section, we first formalize the problem definition. Subsequently, we provide an overview of our approach and describe each step in detail.

Problem Definition. Let I be a scene image and C be the set of known category labels. An object detection network $D(I) = O$, takes I as input and outputs a set of detected objects $O = \{(bb_i, c_i), \ldots, (bb_n, c_n)\}$, where n is the number of detected object instances, bb_i is the bounding box coordinates of the i-th instance, and $c_i \in C$ is the category label of the i-th instance.

A co-occurrence context graph $G = (V, E)$ over a set of scene images $\mathcal{I}$ is a fully connected graph, where a vertex is an object (bb_i, c_i) and the edge between two objects $(bb_i, c_i) \rightarrow (bb_j, c_j)$ encodes the importance of the co-occurrence (i.e., how likely is it that the existence of (bb_i, c_i) can predict the existence of (bb_j, c_j)).

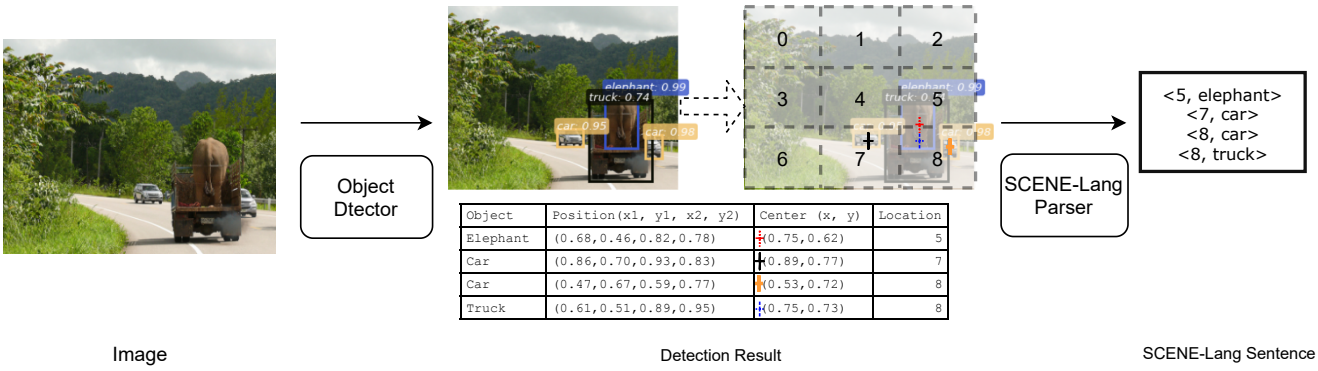
The goal of an attacker is to inject a small perturbation to a scene image $I' \leftarrow I + \Delta I$, so the output of the detection network is manipulated to be $D(I') = O'$ ($O' \neq O$). Our goal is to determine whether a scene image I is adversarial, with the help of the context graph G.

Threat Model. Similar to previous work [21], we assume a strong white-box attack model where attackers have full knowledge of the object detection network $D(\cdot)$. From an attack detection perspective, this provides defense against the strongest possible attack. Previous works [10, 5, 46] have defined three kinds of attacks based on how O' is different from O.

- *Misclassification attack*, where the label of an instance is misclassified, i.e., $c'_i \neq c_i$;
- *Hiding attack*, wherein an instance is not detected by the victim object detector, i.e., $(bb_i, c_i) \notin O'$;
- *Appearing attack*, where an instance that does not exist, is detected by the victim object detector, i.e., $(bb'_i, c'_i) \notin O$.

Overview. Our approach uses context consistency checks to detect adversarial attacks, where the context is defined by the co-occurrence of objects within the scene and their relative positions. The two main challenges in realizing this approach are: (1) how to learn the context graph G (i.e., the edge weights), and (2) given a test time co-occurrence relationship O, how to check whether it is consistent with G.

In this work, we explore the feasibility of using natural language models to solve these challenges. In particular, we first define a new language `SCENE-Lang` to capture the category and coarse-grained location of object instances in a scene image. We can then *describe* the output O of an



Object	Position(x1, y1, x2, y2)	Center (x, y)	Location
Elephant	(0.68,0.46,0.82,0.78)	(0.75,0.62)	5
Car	(0.86,0.70,0.93,0.83)	(0.89,0.77)	7
Car	(0.47,0.67,0.59,0.77)	(0.53,0.72)	8
Truck	(0.61,0.51,0.89,0.95)	(0.75,0.73)	8

Figure 2: A given image is first processed by an Object Detector (e.g. F-RCNN) to get the detection result. As shown in the table, the positions of object's bounding boxes in the detection results are normalized by the width or height of the image. For each object, we compute the center of its bounding box. Then, we map the center to an $H \times W$ grid to get a coarse-grained location. Finally, we convert the processed detection results into a SCENE-Lang sentence. Note that each line in the right-most box is a SCENE-Lang word.

object detection network using a *sentence* in SCENE-Lang. Such scene descriptions (i.e. sentences in SCENE-Lang) form the underpinning for training a language model that essentially models the context graph G. In this work, we use a model based on BERT [7, 27], which we call SCENE-BERT, to learn the intrinsic dependencies between words (i.e., co-occurring objects). Compared to alternative methods like 2-D co-occurrence matrix, graph neural networks, and message-passing-based RNNs, we believe that the attention mechanism of the BERT model allows the capture of dependencies between objects with significantly reduced computation. We also believe that BERT will perform better on naturally occurring, more complicated scenes (i.e., images with more objects) because they can appropriately activate the attention heads from transformers.

During test time, we generate a context consistency score for a scene image I based on the trained SCENE-BERT model. A violation of context consistency in the scene image will lead to a low consistency score. This allows us to detect adversarial attacks by thresholding the consistency score of the composed SCENE-Lang sentence from the scene image. The overall workflow is depicted in Figure 1. We point out here (again) that because SCENE-BERT learns the context from the output of an object detection network $D(\cdot)$, it can work with most detection networks like F-RCNN and YOLO. More importantly, because SCENE-BERT can be trained independently (e.g., using the ground truth labels), applying it to a new detection network neither requires the retraining of the detection network nor SCENE-BERT itself.

3.1. Learning Context with Language Model

SCENE-Lang. We define a new language called SCENE Language (SCENE-Lang) to describe the object co-occurrence information in natural scene images. Each natural scene image can be described with one SCENE-Lang sentence and each word in the sentence is associated with an object instance.

SCENE-Lang Words. We describe the category of an object and its coarse-grained location with a SCENE-Lang word. To describe the location of an object, we evenly divide each image into a $H \times W$ grid and label each grid cell using a number, so we can use a small, finite vocabulary to describe the scene. Using coarse-grained locations can also help tolerate adversarial attacks that may shift the bounding boxes of objects. The center of an object's bounding box determines which cell the object is in. We denote the set location labels as $L = \{1, \ldots, H \times W\}$. Therefore, each SCENE-Lang word $w = (l, c)$ is a pair of a location label ($l \in L$) and a category label ($c \in C$). We denote the finite vocabulary of SCENE-Lang as $W = C \times L$, whose size $|W| = |C| \times H \times W$. Note that although we could encode the object label c_i using a number, because SCENE-Lang is a pseudo language we choose to use the natural language label c_i; this enables ease of explanation upon detecting a context consistency violation.

SCENE-Lang Sentence. We describe the object co-occurrence relationships in a scene image I with a single SCENE-Lang sentence, wherein each word is associated with an object instance in the image. The sentence is represented by $s_I = [w_1, \ldots, w_n]$, where the length of the sentence n is equal to the number of object instances in the image I. We use s instead of s_I subsequently, for ease of exposition. The order of the words in the sentence is sorted based on their location labels (numerically ascending). Figure 2 shows an example about how to describe a scene image with a SCENE-lang sentence.

SCENE-BERT. We use SCENE-BERT, a natural language model to learn the co-occurrence context graph G in natural scene images. Each input to SCENE-BERT is a sequence of tokens, denoted as $\mathbf{T} = [t_1, \ldots, t_n]$. The model also takes a n-dimensional mask vector $\mathbf{M} \in \{0, 1\}^n$ as input, where 0 in the i-th dimension indicates masking off the i-th token t_i and 1 indicates that the corresponding token t_i

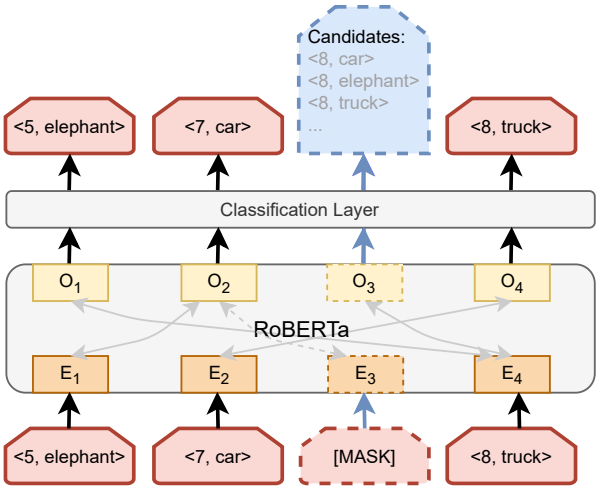


Figure 3: During training or testing time, we selectively mask one or some tokens and ask the SCENE-BERT to predict the masked part. We update weights based on the prediction result at training time. We take the prediction results as the categories, which fit in current context, at test time.

is not masked. The number of tokens n is determined by the number of words in the input SCENE-Lang sentence. The output of SCENE-BERT is a reconstructed sequence of tokens, where the masked off token in $\mathbf{T}$ is replaced with a listed of predicted tokens that match the context (i.e., with lowest cross-entropy loss). We use $f(\mathbf{T}, \mathbf{M}, t_i)$ to denote the confidence score of t_i in the predicted token list. If t_i is not in the list, $f(\mathbf{T}, \mathbf{M}, t_i) = 0$. This score will be used to calculate the consistency score of the whole scene.

SCENE-BERT Architecture. SCENE-BERT is based on the multi-layer bidirectional transformer model BERT [7]. Figure 3 shows the simplified architecture of the model. Since transformers have been widely adopted in language-related tasks and we are reusing an existing implementation of BERT, we omit the detailed description of the model architecture and refer the readers to [40], in the interest of space. Besides being a state-of-the-art language model, we choose BERT to implement our language model for two main reasons. First, the use of the bidirectional self-attention mechanism allows BERT to capture dependencies from both directions (i.e., the prediction of the current word depends on both words appearing before it and after it). This matches our context model very well as the context graph G is not ordered (relationships are both ways). Second, the way BERT is trained is also a perfect match for our task. In particular, BERT is trained with the Masked Language Modeling (MLM) task, where some input tokens are masked at random and the model is asked to predict them. This task is very similar to our approach to detect adversarial attacks (Figure 1): check whether object detection result is consistent with those predicted purely based on context.

Tokenization. Because SCENE-Lang is a pseudo language, tokenizing a sentence s is straightforward. Specifically, we assign each unique word w in the finite vocabulary of SCENE-Lang W, a unique number, which serves at its corresponding token (i.e., $t \in \{1, \ldots, |W|\}$). So the tokenizer simply maps each word w_i in a sentence s to its corresponding number.

Training. We train SCENE-BERT using the same unsupervised masked language modeling task as RoBERTa [27]. Specifically, it randomly masks token(s) from the input sequence, and the objective of the model is to predict the original token(s), only based on the remaining tokens in the sentence (i.e., the context). In other words, SCENE-BERT learns the dependencies between co-occurring objects, or the edge weights in the object co-occurrence graph G. We want to highlight the unique advantage of SCENE-BERT again in that it can be trained with any set of sentences in SCENE-Lang. This means that it can be trained with the ground truth labels of an object detection dataset (as we do in our experiments); this would work with any object detection network trained with the same dataset. Alternatively, it can also be trained in a completely unsupervised manner by running an object detector over a clean dataset to generate the training sentences.

3.2. Checking Context Consistency

In this subsection, we illustrate how we use the trained SCENE-BERT model to perform context consistency checks. At a high level, we use differential analysis to detect inconsistency, i.e., by comparing the detection result (in SCENE-Lang) with the scene description predicted from our context model SCENE-BERT. The smaller the difference, the higher will be the consistency score. Because most adversarial attacks will violate context consistency, by thresholding the consistency score, we can detect whether the input image is adversarial or not. Next, we introduce how we calculate the consistency score.

Let I be a clean scene image and $D(\cdot)$ be the victim object detector. We can encode the output $D(I) = O$ using a SCENE-Lang sentence $s = [w_1, \ldots, w_n]$ as described earlier, which will be tokenized into $\mathbf{T} = [t_1, \ldots, t_n]$. Let $I' \leftarrow I + \Delta I$ be the perturbed adversarial image and $\mathbf{T}'$ be the tokenized SCENE-Lang description over I'. Recall that there are three possible attacking goals, which will affect x in three different ways:

- Misclassification attack where the token associated with an instance is perturbed, i.e., $t'_i \neq t_i$;
- Hiding attack where a token is missing in the token sequence, i.e., $t_i \notin \mathbf{T}'$;
- Appearing attack where an undesired token appears, i.e., $t'_i \notin \mathbf{T}$.

Using the trained SCENE-BERT model, we can mask off a token $t'_i \in \mathbf{T}'$ and ask the model to predict what t'_i is, based on the remaining tokens (i.e., the test time con-

Algorithm 1: Calculate the consistency score of a SCENE-Lang sentence.

Input : Tokenized SCENE-Lang sentence $\mathbf{T} = [t_1, \ldots, t_n]$, the trained SCENE-BERT function $f(\cdot, \cdot, \cdot)$
Output: Consistency score c

$c = 1.0$
$\mathbf{M} = 1^n$
for $i = 1$ *to* n **do**
  $\mathbf{M}[i] = 0$
  $r_{t_i} \leftarrow f(\mathbf{T}, \mathbf{M}, t_i)$
  $c = \min(c, r_{t_i})$
  $\mathbf{M}[i] = 1$
end
return c

text). In theory, if the predicted result is different from t'_i, then we deduce that t'_i is the likely target object under attack. However, this has two associated problems. First, there could be multiple objects that are contextually consistent (i.e., SCENE-BERT can return a list of possible tokens instead of a single one), and hence, how should we calculate the difference between t'_i and the predicted tokens? Second, $\mathbf{T}'$ contains multiple tokens and so, how can we know which token to mask, especially in the case of a hiding attack (where the victim token is missing)? We solve the first problem by using the confidence score of t'_i in the predicted list as the consistency score of that specific object. If t'_i is not in the predicted list from SCENE-BERT, its consistency score will be 0. We solve the second problem by iterating through all tokens (i.e., detected objects) and using the lowest consistency score of all objects as the consistency score of the whole image. The details are captured in Algorithm 1. Note that our approach to calculate the consistency score is able to handle hiding attacks because the missing token typically affects the prediction results of the other non-target tokens.

4. Experimental Analysis

In this section, we evaluate the performance of our approach through comprehensive experiments on two large-scale object detection datasets: PASCAL VOC [9] and MS COCO [24]. We used the two most popular object detection networks: Faster R-CNN [36] and YOLO [35]. We also compare our approach with two state-of-the-art adversarial attack detection methods: a context-agnostic one feature squeeze [44] and another context-aware detection method SCEME [21]. The evaluation includes three types of attacks: misclassification, hiding, and appearing.

4.1. Implementation Details

We use the RoBERTa [27] model, a reproduction of the original BERT model [7], to implement SCENE-BERT. It is configured with six hidden layers and twelve self-attention heads.

Table 1: Attack success rate of three different goals on the PASCAL VOC and MS COCO datasets.

Model	Misclassification	Hiding	Appearing
Results on PASCAL VOC:			
F-RCNN	90.33%	78.09%	96.01%
YOLO	80.16%	89.03%	94.78%
Results on MS COCO:			
F-RCNN	92.78%	82.34%	94.49%
YOLO	79.82%	93.77%	89.74%

The PASCAL VOC dataset contains 20 object categories. The majority of images in the PASCAL VOC dataset have 1 to 5 object instances, on average, 1.4 categories and 2.3 instances per image. The MS COCO dataset contains 80 object categories. Images in this dataset have more object instances, on average, 3.5 categories and 7.7 instances per image. We used the ground truth labels from both datasets to train SCENE-BERT models. Since our context model is designed to consider the co-occurrence consistency of multiple objects in the scene, we omitted images that consist of a single object. For the PASCAL VOC 2007 dataset, we used a 3×3 grid (i.e., $H = 3$ and $W = 3$); so in total we have $|W| = C \times H \times W = 20 \times 3 \times 3 = 180$ tokens. For MS COCO, we also used a 3×3 grid, so in total $|W| = C \times H \times W = 80 \times 3 \times 3 = 720$ tokens.

Since SCENE-BERT can be trained independent of the object detector, we used pre-trained F-RCNN and YOLO models. For the PASCAL dataset, both models were trained with *VOC07trainval* and *VOC12trainval*. For the MS COCO dataset, the F-RCNN model was trained with *coco14train* and *coco14valminusminival*, and the YOLO model was trained with *coco17train*.

To test the attack detection performance, we generate 10,000 attacks for each attacking goal (misclassification, hiding, and appearing) from both datasets, except for hiding attacks on PASCAL VOC, which does not have enough objects for hiding attacks. Because our detection method uses high level semantic information (object co-occurrence context) and does not rely on low-level features, we only evaluate it against digital attacks. The attacks are generated using the standard iterative fast gradient sign method (IFGSM) [20], with $L_\infty \leq 10$ as the perturbation budget; and the perturbations are applied to the whole image. Because SCENE-BERT takes detected object labels and locations as inputs, how the perturbations are generated does not affect the experimental analysis, thus we only used IFGSM. Table 1 shows the attack success rate on the two datasets.

4.2. Baseline Models

We compare our method with two baseline models in the experiments.

Feature Squeeze (FS) [44] is a SOTA context-agnostic method for detecting adversarial image examples. This mechanism can detect the adversarial image examples generated by Fast Gradient Sign Method [14], DeepFool [32], and Projected Gradient Descent [31]. Its core idea is that, adversarial attacks need to limit how many perturbations can be applied (e.g., by limiting the change to the L_2 or L_∞ norm) to achieve (quasi-)imperceptibility. Therefore, by squeezing the input features (i.e., reducing the color bit depth of each pixel and smoothing surrounding pixels), FS may remove enough perturbations and acquire the correct prediction results. Then, by comparing how different the prediction results of the original input and the squeezed input are, FS can detect adversarial attacks.

SCEME [21] is our previous context-consistency-based adversarial attack detection method that showed much better detection performance than Feature Squeeze. It models context at region proposal level and uses attention mechanism and Gated Recurrent Units (GRUs) to learn four types of relationships between region proposals: (1) *spatial context* between regions corresponding to the same object; (2) *object-object context* between regions corresponding to different objects; (3) *object-background context* between regions corresponding to objects and regions corresponding to the background; and (4) *object-scene context* between regions and the whole scene. To detect adversarial attacks, SCEME uses auto-encoders (one per object category) to learn the benign distribution of *context profiles* corresponding to an object category. The context profile contains both edge features and node features (i.e., features of the region proposal). An adversarial attack that violates context consistency will yield a higher reconstruction error rate and by thresholding the reconstruction error rate, SCEME can detect perturbed *regions*. Note that because SCEME works at region proposal level instead of the whole image, we cannot directly compare it with SCENE-BERT. To calculate the detection performance at the whole image level, we aggregate all reconstruction errors from each region proposal and use the highest one as the final score.

4.3. Detection Performance

Evaluation Metric. Given a scene image and an object detector, we aim to determine whether the scene image is adversarial (i.e., the object detector is fooled by the image and makes a wrong prediction). We first compose the SCENE-Lang sentence using the detection result of the scene image output by the object detector. We then use the SCENE-BERT model to calculate the consistency score of the composed SCENE-Lang sentence. We expect that benign/negative images have higher consistency scores and adversarial/positive images have lower consistency scores. By thresholding the consistency score, we are able to plot the receiver operating characteristic (ROC) curve of the detection. We report the area under the curve (AUC) of the ROC curve to evaluate the detection performance.

Table 2: Detection performance for F-RCNN, YOLO on VOC, COCO. (*This configuration is plotted in Figure 4, additional configurations reported in the supplementary material.)

Dataset	Object Detector	Attack Detector	AUC		
			Miscls	Hiding	Appear
VOC	F-RCNN*	SCENE-BERT	0.88	0.74	0.88
		SCEME	0.93	0.95	0.87
		Feature Squeeze	0.53	0.52	0.52
	YOLO	SCENE-BERT	0.89	0.74	0.90
		Feature Squeeze	0.77	0.75	0.79
COCO	F-RCNN	SCENE-BERT	0.84	0.55	0.85
		Feature Squeeze	0.60	0.74	0.60
	YOLO	SCENE-BERT	0.86	0.55	0.88
		Feature Squeeze	0.66	0.60	0.67

Detection Performance. Table 2 shows the detection performance on the PASCAL VOC dataset and the MS COCO dataset. Figure 4 visualizes the AUC curves on the PASCAL VOC dataset with F-RCNN under different attack setups for better comparison with SCEME and FS. Overall, SCEME and SCENE-BERT, both of which are the context-aware detection methods, significantly outperformed Feature Squeeze, which is a context-agnostic method. The only exception is hiding attacks on the MS COCO dataset. The reason is that images from the MS COCO dataset have more objects, so hiding a single object usually will not significantly reduce the context consistency. We believe the results once again validate the effectiveness of context consistency-based detection approach. Comparing SCEME and SCENE-BERT, we observed that SCEME still outperformed SCENE-BERT. We attribute this to the richer features used by SCEME (e.g., object-background and object-scene context). However, SCENE-BERT also has its advantages over SCEME. First, SCENE-BERT is model-agnostic, so we can also pair it with YOLO without any modification to YOLO or retraining; on the other hand, SCEME is tightly coupled with the Faster R-CNN architecture. Second, SCENE-BERT is also faster as it only iterates through detected object instances, whereas SCEME needs to iterate through hundreds of region proposals.

Effectiveness of Locations. To understand the importance of the coarse-grained location feature in our approach, we also performed the attack detection task with a relaxed consistency check, where we only check the category and ignore the location when calculating the consistency score. We name this approach SCENE-BERT Relax and the full version SCENE-BERT Strict. The results are shown in Figure 5. As we can see, the AUC is higher across all three types of attacks when we also check the coarse-grained location when calculating the consistency score. We believe this shows (1) SCENE-BERT is able to capture location related dependencies between objects, and (2) even coarse-grained location information can help better detect the attacks.

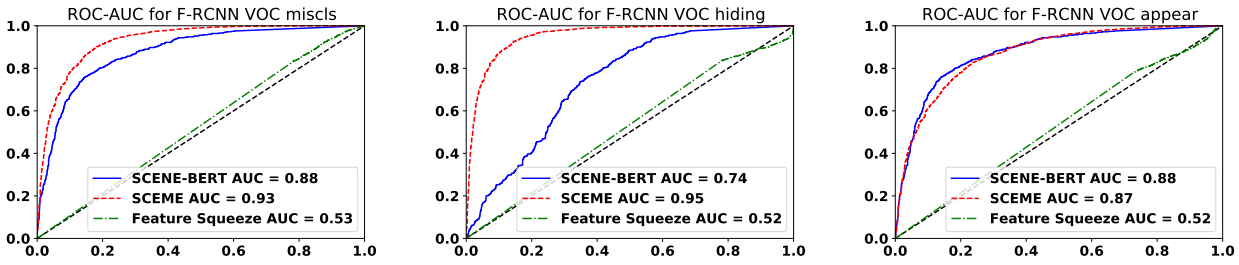


Figure 4: Detection performance on the PASCAL VOC dataset.

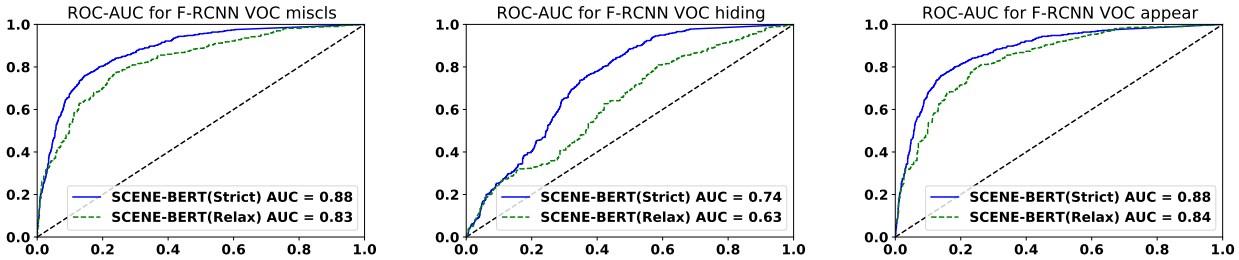


Figure 5: SCENE-BERT Strict vs. SCENE-BERT Relax on PASCAL VOC.

4.4. Case Studies

While SCENE-BERT performed slightly worse than SCEME over the datasets, we also observed cases where SCENE-BERT can detect attacks that SCEME cannot. Figure 6 shows two cases. In the first case, the left bird (peacock) is perturbed to be a boat; and in the second case, the left horse is perturbed to be a dining table. Both cases obviously violate the context consistency based on object co-occurrence hence were detected by SCENE-BERT. We suspect that the reason that SCEME did not detect these attacks is because it also considers the visual features of the object and the auto-encoders may focus more on the visual features instead of the context.

While analyzing the evaluation results, we also noticed that if the attack is context consistent (e.g., misclassifies a bus to a car), then SCENE-BERT cannot detect such attacks. We want to argue that context consistency is one way to check for an attack and need not supplant, but can complement, other methods that can check individual objects or removal/addition of objects. Moreover, such context consistent attacks are likely to be less disruptive. For example, mis-clsassifying a bus to a car is unlikely to cause an autonomous vehicle to collide with the bus, but changing a speed limit sign in the middle of a road to a stop sign can lead to disastrous outcomes.

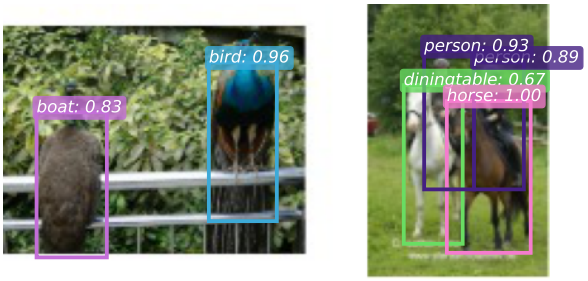


Figure 6: Examples where SCENE-BERT is able to detect the attack but SCEME does not.

5. Conclusion

Motivated by the observation that language descriptions of a natural scene images have captured the object co-occurrence relationship, we propose using a language model to learn the dependencies between objects and using the trained model to perform context consistency checks to detect adversarial attacks. Compared to previous context-consistency-based detection method, our approach can be paired with most object detectors and does not require modification or retraining to the object detector. Our experiments show that our method is very effective in detecting a variety of attacks on two large scale datasets: it significantly outperforms a state-of-the-art context-agnostic method and is comparable to previous context-aware method that is model-dependent.

Acknowledgments. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00112090096. Approved for public release; distribution is unlimited.

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 274–283. PMLR, 2018. 1, 3

[2] Ehud Barnea and Ohad Ben-Shahar. Exploring the bounds of the utility of context for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7412–7420. IEEE, 2019. 2

[3] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z Morley Mao. Adversarial sensor attack on lidar-based perception in autonomous driving. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 2267–2281. ACM, 2019. 3

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017. 1, 3

[5] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Polo Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD*, pages 52–68. Springer, 2018. 3

[6] Myung Jin Choi, Antonio Torralba, and Alan S Willsky. A tree-based context model for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):240–252, 2011. 2

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 4171–4186, 2019. 4, 5, 6

[8] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 364–380. Springer, 2018. 2

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 2, 6

[10] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Physical adversarial examples for object detectors. In *Proceedings of the USENIX Workshop on Offensive Technologies (WOOT*, 2018. 1, 3

[11] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017. 3

[12] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2009. 2

[13] Ian Goodfellow, Patrick McDaniel, and Nicolas Papernot. Making machine learning robust against adversarial inputs. *Communications of the ACM*, 61(7):56–66, 2018. 1, 3

[14] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 1, 3, 7

[15] Abhinav Gupta and Larry S Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 16–29. Springer, 2008. 2

[16] Dan Hendrycks and Kevin Gimpel. Early methods for detecting adversarial images. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 3

[17] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3588–3597. IEEE, 2018. 2

[18] Xiaojun Jia, Xingxing Wei, and Xiaochun Cao. Identifying and resisting adversarial videos using temporal consistency. *arXiv preprint arXiv:1909.04837*, 2019. 1

[19] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1219–1228. IEEE, 2018. 2

[20] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 1, 3, 6

[21] Shasha Li, Shitong Zhu, Sudipta Paul, Amit Roy-Chowdhury, Chengyu Song, Srikanth Krishnamurthy, Anantharam Swami, and Kevin S Chan. Connecting the dots: Detecting adversarial perturbations using context inconsistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 396–413. Springer, 2020. 1, 2, 3, 6, 7

[22] Bin Liang, Hongcheng Li, Miaoqiang Su, Xirong Li, Wenchang Shi, and Xiaofeng Wang. Detecting adversarial image examples in deep neural networks with adaptive noise reduction. *IEEE Transactions on Dependable and Secure Computing*, 2018. 3

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988. IEEE, 2017. 2

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 2, 6

[25] Jiayang Liu, Weiming Zhang, Yiwei Zhang, Dongdong Hou, Yujia Liu, Hongyue Zha, and Nenghai Yu. Detection based defense against adversarial examples from the steganalysis point of view. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4825–4834. IEEE, 2019. 3

[26] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 21–37. Springer, 2016. 2

[27] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 4, 5, 6

[28] Yong Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Structure inference net: Object detection using scene-level context and instance-level relationships. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6985–6994. IEEE, 2018. 2

[29] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 852–869. Springer, 2016. 2

[30] Pingchuan Ma, Stavros Petridis, and Maja Pantic. Detecting adversarial attacks on audiovisual speech recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 6403–6407. IEEE, 2021. 3

[31] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 1, 3, 7

[32] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582. IEEE, 2016. 7

[33] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 891–898. IEEE, 2014. 2

[34] Aude Oliva, Antonio Torralba, Monica S Castelhano, and John M Henderson. Top-down control of visual attention in object detection. In *Proceedings the International Conference on Image Processing*, volume 1, pages I–253. IEEE, 2003. 2

[35] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788. IEEE, 2016. 1, 2, 6

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 91–99, 2015. 1, 2, 6

[37] Mohammad Amin Sadeghi and Ali Farhadi. Recognition using visual phrases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1745–1752. IEEE, 2011. 2

[38] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014. 1, 3

[39] Antonio Torralba. Contextual priming for object detection. *International Journal of Computer Vision*, 53(2):169–191, 2003. 2

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008, 2017. 5

[41] Chaowei Xiao, Ruizhi Deng, Bo Li, Taesung Lee, Benjamin Edwards, Jinfeng Yi, Dawn Song, Mingyan Liu, and Ian Molloy. Advit: Adversarial frames identifier based on temporal consistency in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3968–3977. IEEE, 2019. 1, 3

[42] Chaowei Xiao, Ruizhi Deng, Bo Li, Fisher Yu, Mingyan Liu, and Dawn Song. Characterizing adversarial examples based on spatial consistency information for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 217–234, 2018. 3

[43] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5410–5419. IEEE, 2017. 2

[44] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2018. 2, 3, 6, 7

[45] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 684–699. Springer, 2018. 2

[46] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1989–2004. ACM, 2019. 3

[47] C Lawrence Zitnick, Devi Parikh, and Lucy Vanderwende. Learning the visual interpretation of sentences. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1681–1688. IEEE, 2013. 2